ANALYSIS OF NETWORK PROPERTIES USING SELF ORGANIZING MAPS FOR SERVICE DEPLOYMENT ON THE CLOUD

by

Emel Uras Balkanli

Submitted in partial fulfillment of the
requirements for the degree of
Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
August 2017

I dedicate this thesis to my loving parents, Zekiye Uras, Selim Uras, Gulay Balkanli and A. Erdal Balkanli who have encouraged me to study Master of Computer Science and supported me throughout the process. I also dedicate this study to the most special person in my life, Eray Balkanli, who have kept me motivated for all the time and never left my side.

Table of Contents

List of Tables

List of Figures

Abstract

This work offers in-depth analysis of network properties to employ them for service deployment on cloud systems. The proposed analysis is evaluated on three different data sets from different locations, captured in 2012, 2013 and 2014 to provide insights into network properties and how to use them while deploying services. This research proposes the employment of a Self-Organizing Map as a type of Artificial Neural Network that generates a low-dimensional representation of high dimensional data using unsupervised learning methods. My analysis shows that there are significant effects of selected network properties, namely latency, success status, hop count and time-to-live, on the optimal location for the service to be deployed. In summary, using the proposed technique for analysis of network properties to choose the location for service deployment on the cloud could help to understand where to deploy the service to increase efficiency with respect to the selected properties.

List of Abbreviations Used

ASR Average Success Rate.

CAIDA Cooperative Association for Internet Data Analysis.

EM Expectation Maximization.

ICMP Internet Control Message Protocol.

IP Internet Protocol.

IPv4 Internet Protocol Version 4.

JSON JavaScript Object Notation.

MATLAB Matrix Laboratory.

MSS Mean Sum of Squares.

NN Neural Network.

RTT Round Trip Time.

SOM Self Organizing Maps.

TPH Time per Hop.

TTL Time-to-Live.

WEKA Waikato Environment for Knowledge Analysis.

Acknowledgements

First of all, I would like to thank my supervisor, Dr. A. Nur Zincir-Heywood, for her continuous support, guidance and encouragement. The door to her office was always open whenever I ran into a trouble or had a question about my research or writing. Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Malcolm Heywood, Dr. Andrew McIntyre, and Dr. Kirstie Hawkey for their encouragement and insightful comments. Moreover, there are no words to express how grateful I am to my family and husband, Eray Balkanli, for always supporting, helping and motivating me to study hard and finalize this research.
This research is conducted as a part of the Dalhousie NIMS Lab:

https://projects.cs.dal.ca/projectx/

Chapter 1

Introduction

From past to present, efficiency is one of the most popular issue of computer system. Although a series of revolutions in computer systems has been occurred, efficiency is still an issue because of the increment of users, transactions and computational complexity. Today, around 40% of the world population has Internet connection and 45,549 GB of Internet traffic occurs in 1 second[1]. For this reason, as the hardware or device abilities are increased, the network performance may not allow to use the hardware features as expected. At this point, to locate the machines / servers in optimal locations with respect to network properties (metrics) of the current network infrastructure may increase the efficiency of the whole system. Location optimization is to find an optimal location that has the most cost effective or highest achievable performance under the given constraints, by maximizing desired factors and minimizing undesired ones. Thus, the service location optimization on the cloud systems provides to serve better, using less servers and / or using the current servers more effectively depending on the objective(s) of the service provider on the cloud.

Service (server) allocation problem is a popular research area on cloud, especially for the network systems. Even though locating multiple servers is a potential solution to manage large networks, this is very costly. Therefore, many network managers tend to use fewer servers while minimizing the travel time as well as the average waiting time for the transferring of data in the large networks. Having a single server in the most efficient location is one of the cheap solutions, which also provides minimal configuration and easiness in maintenance. On the other hand, besides being risky against system break downs, it is critical to point out the most optimal location for the server. It is important to emphasize that an analysis on a large data collected on different points by real users over the network is required to reveal the points that have maximum traffic and their characterization. My purpose is to present a novel approach to explore the most optimal location(s) for deploying a service to control

and handle all its related network requirements. In short, I aim to shed light into the following questions during my research:

- What is the best way to find out the most optimal location to place a single server to manage a large network with the maximum efficiency for a selected service? What are the advantages and disadvantages of such an analysis?
- How the service data is analyzed - how to analyze it and to highlight its major trends? What are the main challenges of suggested method related with big data analysis?
- What type of machine learning based algorithms should be employed for optimization? Is it possible to achieve high accuracy in categorizing large amount of network traffic via clustering approaches? Which features in such traffic are top informative?

To study the above research questions, I begin with analyzing the main characteristics of the network traffic by measuring the reliability, cost and performance of packet delivery in order to understand its nature. To this end, I employ 9 publicly available datasets collected by Ark monitors [5] by The Cooperative Association for Internet Data Analysis (CAIDA). CAIDA is an independent research group based at the University of California, San Diego. CAIDA has been collecting the datasets since 2008 from all over the world and provides a huge source to analyze the characterization of the Internet based on locations and time periods. I focused three major continents, namely Asia Pacific, Europe and North America, since those locations contain the highest number of monitors set up by CAIDA, for the years 2012, 2013 and 2014. Note that I have selected these years to observe changing trends for internet data, and the dataset belongs to 2014 is the most recent, fully completed and available one collected by Ark Monitors provided by CAIDA.

To find optimal location for servers, specific data features such as latency, hop count, probe and reply TTL are selected. I focus on the reliability and robustness of the network packets and the performance of the network. In order to characterize the performance, it is required to measure Round-Trip-Time (RTT) values between pairs of hosts [26]. Moreover, I aim to clarify the effect of the relation between the selected features on the success of packet delivery.

In addition, to investigate the optimal location, two different unsupervised learning algorithms are explored, namely Self Organizing Maps (SOM) and K-means. The purpose of using these clustering methods is to take an abstract glance at data, and then develop some logical structures based on characteristics of groups before going deeper into the detailed analysis. SOM is used because of the low-dimensional map it creates as an output. This allows to decide the optimal location just by looking at the map visualization. On the other hand, K-means is an alternative method which is applied to compare the results with SOM and emphasize the possible differences based on the clustering results from both algorithms. Venkatkumar et al. reveals that the K-Means algorithm is faster and also produces quality clusters when using huge datasets in comparison with Hierarchical Clustering, DB Scan Clustering, Density Based Clustering, OPTICS and EM Algorithm [31]. This explains the reason of employing the K-means algorithm as the second clustering method instead of the clustering algorithms mentioned above.

Finally, I employ the tools Matlab and Weka for running SOM and K-means algorithms, respectively. Thus, I aim to point out the main differences between those tools, where Matlab is a commercial software while Weka is an open source, on the data analysis and pre-processing steps.

The rest of the thesis is organized as the following: Chapter 2 discusses the related work in this field. Chapter 3 introduces the datasets, techniques, tools and performance metrics employed in this thesis. Chapter 4 presents the evaluation results, the benefits and the limitations of the proposed methodology. Finally, Chapter 5 draws conclusions and discusses the future work.

Chapter 2

Literature Review

In this section, the review of related works in the area of server location optimization for service deployment purposes in the literature are presented . Berman et al. worked on optimizing server location in [11] by designing a sequence strategy by using queuing systems for the incoming network packets to a server via median-based solutions. They propose two methods to find optimal server location: i) Selecting the location with minimal cost and average weighted travel time, ii) Selecting the location with the lowest travel time and average cost. Engel et al. revealed in [15] that load-balanced locations provide a secure and reliable solution in the long term. They proposed a scheme to limit IP layer processing and find the optimal location for a network by controlling the load distribution of each potential location.

Wang et al. explored the importance of co-locating production servers at the Internet exchange points and focused on locating satellite data centers on the most optimal areas via measure-oriented techniques [32]. They developed a specific Javascript-based tool, called CloudBeacon, to collect data over Internet and to measure the delay and the throughput from the users to their target hosts. Then, they Reflection Ping measurement method to analyze the performance of combined large-scale infrastructures. They revealed that the latency is the lowest in North America while Asia Pacific and Europe experience high latency, where more than 72 locations inside those continents were analyzed. Larumbe et al. revealed that the location of the data centers, servers and the traffic routing between the software components are extremely crucial in overall network performance over an in-depth literature survey [23]. Then, they provided a mathematical framework for the issue: Cloud Location and Routing, which presents the factors affected by server locations and enables one to calculate the data center costs easier. They also demonstrated that a balanced distribution of data centers is able to reduce the average and propagation delay. Jung et al. focused on decreasing the response time of virtual machines in a data center, and

they proposed an approach that dynamically allocate the online resources based on the location of users and data centers on cloud computing environments [18]. There are not much information about the data the researches employed but 10 data centers including 25 user requests each over 10 different zones. They stated that their model enables one to utilize the data center at maximum while the performance of the machines in the data center stays stable. Nygren et al. researched on Akamai Network, which has more than 61,000 server over 70 countries, after exploring the main Internet application requirements and delivery challenges to observe how Akamai overcomes those drawbacks [27]. From my perspective, they found out that Akamai locates their servers strategically to provide closer distances between their servers and the end-users to keep the delay on the network at the lowest level. Zegura et al. presented an approach to estimate the response time of a client to access a server, and they used the location information as their one of the main dependants [28]. They specifically selected 4 servers, one in Los Angeles, one in Washington and two in Atlanta, and 20 clients, four in Maryland and demonstrated that sixteen in Atlanta. Then, they figured out the number of hops between the servers and the clients and used three server location algorithm to reach the optimal performance of the network. They discovered the importance of server location in a network and demonstrated that their approach gives the best results than the Nearest Server Selection and Random Server Selection algorithms. Last but not least, Berman et al. discovered a novel solution for multiple server location problem by analyzing the user demands on each node of a network based on requests sent in addition to average travel time of network packets from a server to its closest clients to minimize the travel time in [10].

Aforementioned studies mostly provide approaches to optimize server location for a network. My research is complementary to these studies in terms of aiming to find out the best potential point to set up a main server for a large network. My research is novel since I employ an unsupervised machine learning approaches such as SOM and K-means with specifically selected set of features, latency, failure, hop count, probe TTL and reply TTL, to perform such an analysis. To the best of my knowledge, this is the first work analyzing the performance of the selected unsupervised learning algorithms on the server location optimization for service deployment. Employing publicly available data makes this research to be easily validated and repeatable.

Chapter 3

Methodology

This chapter presents the main characteristics of the employed network dataset, the description of employed mathematical computation and machine learning environment systems, and the focused unsupervised learning systems used as well as the selection of optimization criteria metrics and performance analysis applied in this thesis.

3.1 Dataset Characterization

I employ publicly available IPv4 Routed /24 Topology datasets for this research. These datasets are presented by CAIDA. They include forwardIP path data collected by examining random /24 prefixed IP addresses on the Internet to reveal Internet topology characterization and to measure the potential latency between specific locations. The data are collected via 181 different monitors in 60 unique countries from 2008 to present, shown in the figure 3.1.

In this experiment, three main geographical regions are selected to apply the proposed method. These regions are: North America, Asia Pacific and Europe. In addition, three sub-locations are selected for each region. The selected locations in

Figure 3.1: CAIDA Dataset Map

Table 3.1: Data Set Summary Table

Features Employed		Data Sets Employed		
		N. America	Europe	Asia Pacific
Latency (ms)	<25	58.4%	44.1%	31%
	25-50	27.8%	32%	37.6%
	51-75	9.3%	14.6%	19.8%
	76-100	2.2%	5.6%	7.8%
	>100	2.3%	3.5%	3.5%
# of Hop Counts	<10	10.9%	2.4%	2.2%
	10-20	80.6%	56.4%	70.8%
	21-30	8.4%	38.6%	26%
	>30	0.1%	2.6%	1%
Probe TTL	<=5	81.1%	45%	54.7%
	>5	18.9%	55%	45.3%
Reply TTL	<100	14.4%	1.9%	1.1%
	100-200	75.6%	87.9%	83.2%
	>200	10%	10.2%	15.7%
Locations (where the dataset is captured)		San Jose Ottawa Virginia	Ireland Germany England	South Korea Sydney Tokyo

North America are Virginia, San Jose and Ottawa. Japan, South Korea and Australia are selected as Asia Pacific sub-locations. Lastly, Ireland, Germany and England are selected from Europe. The approach proposed is that for a given region, a number of (sub) locations from the region can be chosen (based on the objectives and / or policies of the service provider) as potential locations to deploy the servers for a given service. Then the unsupervised learning algorithm can be applied to suggest potential optimal locations to the human expert. Finally, human expert selects the best one based on the policies of the service provider. The optimal locations that are found for each region by the unsupervised learning algorithm can be compared among each other and the overall optimal location can be found. The data selection is not only based on the locations but also the date of the data gathered. The reason behind this is that the time specific situations on network might affect the optimization results. For instance, the network is expected to be used by less number of users and the performance could be better during the holidays. For this reason, the selected dates are not a holiday for selected locations. The dates 22 January, 22 April, 22 July and 22 October are selected from each location.

```
{"version":"0.1","type":"trace", "userid":0, "method":"icmp-echo-paris", "src":"205.189.33.78",
"dst":"91.222.160.129", "icmp_sum":33306, "stop_reason":"COMPLETED", "stop_data":0, "start":
{"sec":1388627162, "usec":535331, "ftime":"2014-01-01 21:46:02"}, "hop_count":6, "attempts":3,
"hoplimit":0, "firsthop":1, "wait":5, "wait_probe":0, "tos":0, "probe_size":44, "hops":[
{"addr":"205.189.33.1", "probe_ttl":1, "probe_id":1, "probe_size":44, "rtt":7.731, "reply_ttl":254, "reply_tos":0,
 "reply_size":56, "reply_ipid":0, "icmp_type":11, "icmp_code":0, "icmp_q_ttl":1,  "icmp_q_ipl":44,"icmp_q_tos":0},
{"addr":"66.244.255.93", "probe_ttl":2, "probe_id":1, "probe_size":44, "rtt":7.908, "reply_ttl":254, "reply_tos":192,
 "reply_size":56, "reply_ipid":1307, "icmp_type":11, "icmp_code":0, "icmp_q_ttl":1, "icmp_q_ipl":44, "icmp_q_tos":0},
{"addr":"66.163.66.141", "probe_ttl":3, "probe_id":1, "probe_size":44, "rtt":8.282, "reply_ttl":253, "reply_tos":0,
 "reply_size":96, "reply_ipid":53408, "icmp_type":11, "icmp_code":0, "icmp_q_ttl":1, "icmp_q_ipl":44, "icmp_q_tos":0},
{"addr":"66.163.75.85", "probe_ttl":4, "probe_id":1, "probe_size":44, "rtt":18.645, "reply_ttl":252, "reply_tos":0,
 "reply_size":96, "reply_ipid":40211, "icmp_type":11, "icmp_code":0, "icmp_q_ttl":1, "icmp_q_ipl":44, "icmp_q_tos":0},
{"addr":"66.163.65.114", "probe_ttl":5, "probe_id":1, "probe_size":44, "rtt":18.837, "reply_ttl":251, "reply_tos":0,
 "reply_size":96, "reply_ipid":21026, "icmp_type":11, "icmp_code":0, "icmp_q_ttl":1, "icmp_q_ipl":44, "icmp_q_tos":0},
{"addr":"62.115.12.85", "probe_ttl":6, "probe_id":1, "probe_size":44, "rtt":18.708, "reply_ttl":250, "reply_tos":0,
 "reply_size":56, "reply_ipid":0, "icmp_type":11, "icmp_code":0, "icmp_q_ttl":1, "icmp_q_ipl":44, "icmp_q_tos":0}]}
```


Figure 3.2: Json data example

As is mentioned before, the data have been collected for nine years by CAIDA, so the datasets are huge! Therefore, I applied a sampling method to have reasonable amount of data. By doing so, I aim to arrange the input sets based on the systematic sampling technique, which is explained in detail in [12]. The reason of using systematic sampling instead of using any other sampling methods is to eliminate the possibility of having the non-reliable data regarding to location and time. For example, if any sampling method, which is based on randomness, is applied, there is a chance to select the data for a holiday, non-existence date, the dates that are close to each other or the locations that are geographically close to each other. It should be noted here, there are some missing dates for some locations in CAIDA dataset. Shortly, such subsets including equal population size are created via circular processes returning from the end to the beginning after each round over the complete data. An element $E(k)$ is randomly selected from a subset, then every k th element is selected from each subset, where k refers to the sampling interval which is equal to the total size over the subset size. Table **??** presents a summary of the statistical properties of the data sets employed.

The dataset is collected using packet prober called Scamper that is designed by CAIDA to actively probe destinations in a timely fashion [25]. Warts is the default data format of Scamper that contains crucial meta data regarding each individual packet [8]. Scamper includes an API that allows its binary output files to be easily read. However, I converted the data into another format to be able to analyze the output not only in Scamper but also in different platforms. Scamper API also helps to convert data into pcap, json, text, cat and dump formats. I convert the data from warts to json which is a structural, language-independent, text-based data format

Table 3.2: Example Data Format after Feature Selection

Failure	Hop count	RTT	Probe TTL	Reply TTL	Class
4	12	14.5283	12.0	175.0	1
3	15	22.5219	12.0	142.0	1
4	28	22.7129	23.0	201.0	1
1	17	77.7292	17.0	179.0	1
1	22	52.2559	22.0	182.0	1

that is easy for humans and machines to read and write [4]. Moreover, I created a Java application for feature selection. Java contains json library that makes the application more efficient. Figure 3.2 shows a json output for one probe.

As you can see in the figure 3.2, there is general information about the packet and also detailed information about each hop that the packet travels. The probe size is fixed for all probes in a given dataset and defined as 44, that is why the probe size is not selected as an attribute. STOP-REASON indicates the failure-success situation of the packet. 'Failure' has four different values as COMPLETED, LOOP, GAPLIMIT and UNREACHED. The values are labeled as decimal numbers respectively 1, 2, 3 and 4. 'Hop-count' is the number of hops that the packet travels until reaching the destination or dropping by a reason. 'Probe TTL' (Time To Live) increases after each travelled hop, I took into consideration the probe TTL value at the last hop for each packet. 'RTT' (Round Trip Time) is another selected attribute that one of the most crucial network performance criteria. A Java application is written to select the attributes and eliminate the rest while using JSON library. The example output of the java application is shown in the table 3.2. At this point, the data can be imported as matrix into Matlab and Weka and can directly be used for data analysis and unsupervised learning, i.e. clustering, algorithms.

The data amounts are large enough to represent most of the individuals, thus findings can be generalized to the population. The table 3.3 shows the number of packets and the size of employed datasets. Data sizes are around 7-10 GB before the feature selection application is applied. The java application eliminates the features that are not selected for this study and changes format into column-row representation. Column-row format provides flexibility to make different calculations on every metrics. Also, the file size is decreased after feature selection which makes the process

Table 3.3: Sizes of the Datasets

	2012			2013			2014		
Location	Asia Pacific	Europe	N. America	Asia Pacific	Europe	N. America	Asia Pacific	Europe	N. America
Json File Size (GB)	7.8	9.4	8.9	9.8	10.4	9.2	7.1	7.6	7.0
.dat File Size (MB)	80.5	90.3	112.7	100.9	103.1	112.3	74.5	76.2	84.9
# of packets	2,374,221	2,657,465	3,394,095	2,969,800	3,030,771	3,377,067	2,192,319	2,251,364	2,535,432

faster.

In my opinion, the results would be promising since they would contribute to the literature although they are opposite to the expected ones. Also, the dataset reflects a large miscellany of traceroute measurements and ensures this research to be easily validated and compared to the other researches in this field. If the results are as expected in the end, it would demonstrate that deploying a server in an efficient location for large networks provides network administrators the possibility of managing the whole network with maximum success rate and minimum travel / waiting time.

3.2 Learning Systems Employed

This section describes the optimization and clustering systems employed in this thesis to explore a method based on clustering with respect to the network performance metrics.

3.2.1 Employeed Libraries in JAVA

JSON is a structural data format that uses the JavaScript syntax for describing data objects. Also, Json is a platform independent language and there are many parsers and libraries have been developed over the years. Java basically has 7 different libraries that provides to reading a JSON file, convert JSON objects to Java format, save it to a file and backwards. json.simple and jackson libraries are used to parse json files. To convert JSON attributes to Java objects provides great convenience to make calculations.

3.2.2 MATLAB

Matlab is a platform that provides environment for the solution of many scientific and engineering problems. Also, Matlab provides environments for both coding and graphical interface to apply solutions even without having coding experience. Matlab

has a strong visualization representation to gain insights from data and interpret the result easier. Matlab is an interactive system whose basic data element is an array that does not require dimensioning.

The name Matlab stands for matrix laboratory. Matlab was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects, which together represent the state-of-the-art in software for matrix computation.

Matlab has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, Matlab is the tool of choice for high-productivity research, development, and analysis.

Matlab features a family of application-specific solutions called toolboxes. Very important to most users of Matlab, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of Matlab functions (M-files) that extend the Matlab environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, optimization, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

In this research, Self-Organizing Map data-analysis method is employed. The method produces low-dimensional mapping of high-dimensional data distributions, with respect to the similarity relations between the data items. Matlab has a toolbox that is available as a set of SOM functions programmed and also to write the scripts that use these functions to implement a specific SOM algorithm. The essential parts for application of SOM are prepossessing of the input data and selecting appropriate inputs for function parameters, in order to achieve reliable results. The selection affects both the reliability of the results and the efficiency of the experiment. For instance, if the dimension size is selected as a small value, the results will be obtained faster but the margin of error will be higher because of having less data dispersion. Thus, the dimension size ought to be selected by considering the threshold.

The figures 3.3 are the examples of n x m mapping representations of Matlab SOM toolbox. Note that, n and m stand for row-column sizes and the total cluster size is the multiplication of the row and column size of the map, which is the result of n x m. The row-column sizes might be equal as in the example figure. Also, the

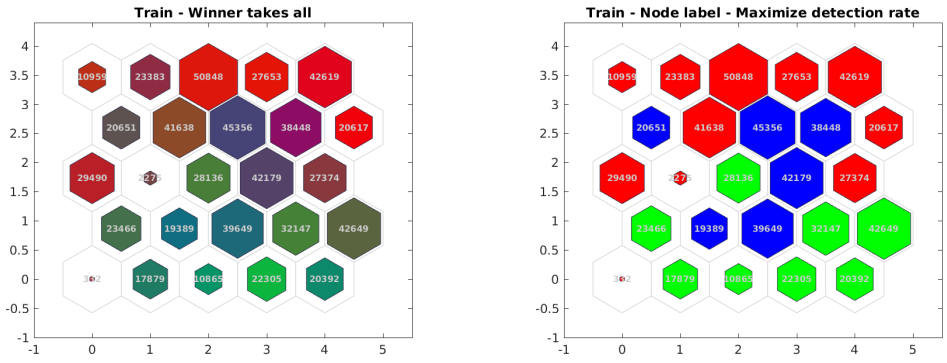

Figure 3.3: Example SOM results

data is represented by colors, different size of colored area and the textual labeling on each hexagonal zones according to the classes of input data items mapped onto the SOM. Each representation gives different information about the data dispersion and the clusters. Basically, the nodes are purely painted by three basic colors, which are red, green, and blue or painted by the mixture of the three colors regarding to the intensities of the corresponding classes belonging to the same node. In another words, there exists an indefinite number of colors in practice. Class1, class2 and class3 match to the colors red, green and blue, respectively. If one of the class size is considerably larger than the others, the color of node becomes purer. Otherwise, the color will be the mixture of two or three colors. The colored area size shows the total amount of data grouped into cluster. In the same way, the label on each cluster is the total number of instances.

Each location corresponds to one class in the experiment and each class is expressed as a different color. The first map of figure 3.3 shows the nodes painted as the different intensity of colors. The map gives an idea about the data amounts of each class located into each node. For instance, the third node in the first row mostly includes the class1 data, because it is represented by a color which is a similar color to the pure red. However, the second node in the first row contains mostly class1 and class3 data, since the purple color is the mixture of red and blue. If I only consider the most frequently located classes for each node, it is possible to produce an output that has the nodes colored by the color of majority class. The second map of the figure 3.3 is an example of the representation of the majority classes. In this case, the map does not give an information about the intensities of the classes and there

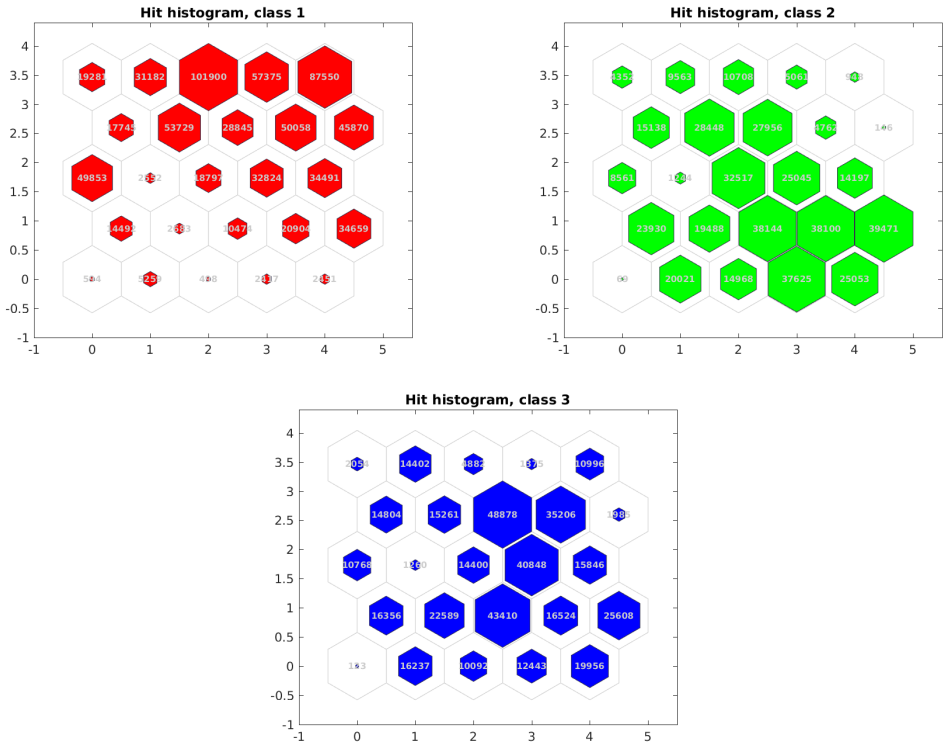

Figure 3.4: SOM output maps

are only three possible colors for each node which are red, green and blue.

It is also possible to see the dispersion of each class onto SOM map, separately. The figure 3.4 shows the individual class maps. The maps can be used to interpret the distribution of the classes. For example, class2 has more bigger size colored nodes, which means there is a strong tendency for the data to take on central values and low probability of large deviation for this class. If the deviation is the decision criteria of a problem, considering the individual class representation is more useful.

3.2.3 WEKA

Weka (The Waikato Environment for Knowledge Analysis) is an open-source software that provides solutions for data mining problems. Weka has an API that supports to add new Java classes to solve specific problems. It also has a user interface to apply the algorithms directly. The last but not the least, Weka has a command line interface that allows to use all features of the software and is very useful for scripting large jobs. Moreover, Weka is platform independent which provides flexibility about

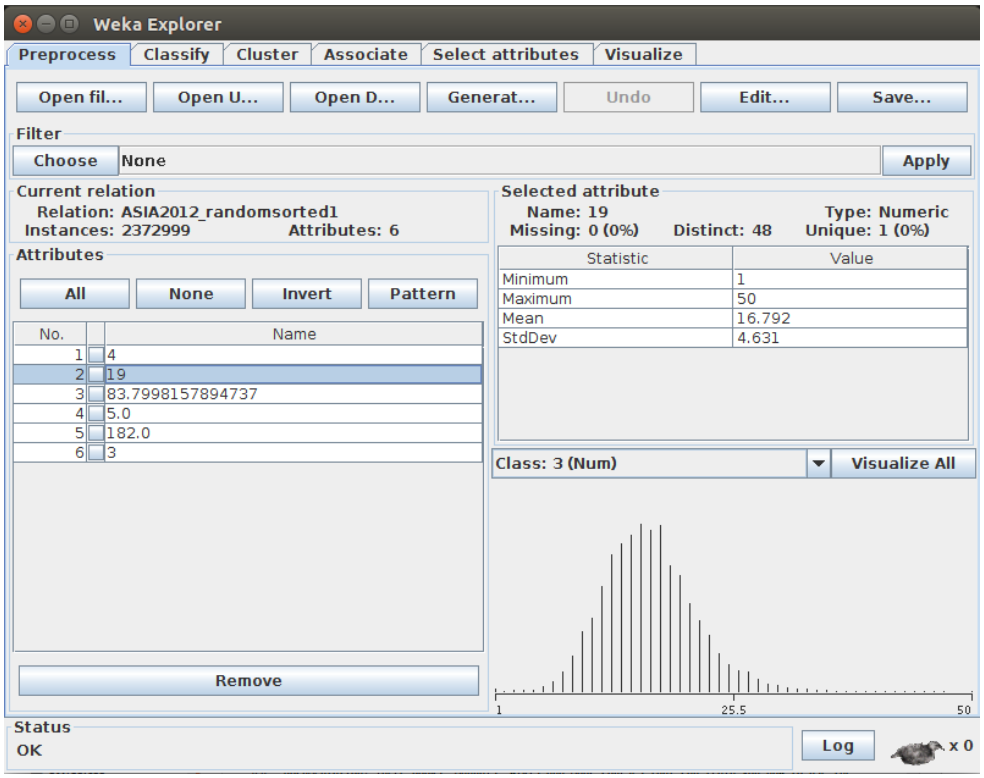

Figure 3.5: Weka Explorer Page

working environment. Weka input data format is .arff which consist of a header and data section parts.

Weka can be used not only for application of data mining algorithms but also for data pre-processing, regression, classification, clustering, association rules,cross-validation and visualization. The figure 3.5 indicates the explorer page of Weka. As is seen, Along with the options to edit the data, it is possible to gain insight about the data such as minimum, maximum values and the distribution of the attributes. There are wide range of documentation that can help to train people how to use the platform effectively.

Weka is used to apply K-means clustering algorithm for this study. Weka provides detailed information for each cluster. Also, data can be visualized in coordinate plane and x and y axes can be selected as one of the data attributes, data instance number or cluster number. Since, both the distribution of data, according the two selected attributes as x and y coordinates and also the distribution of the data with respect to belonging clusters can be seen. Although the method provides more detailed visual

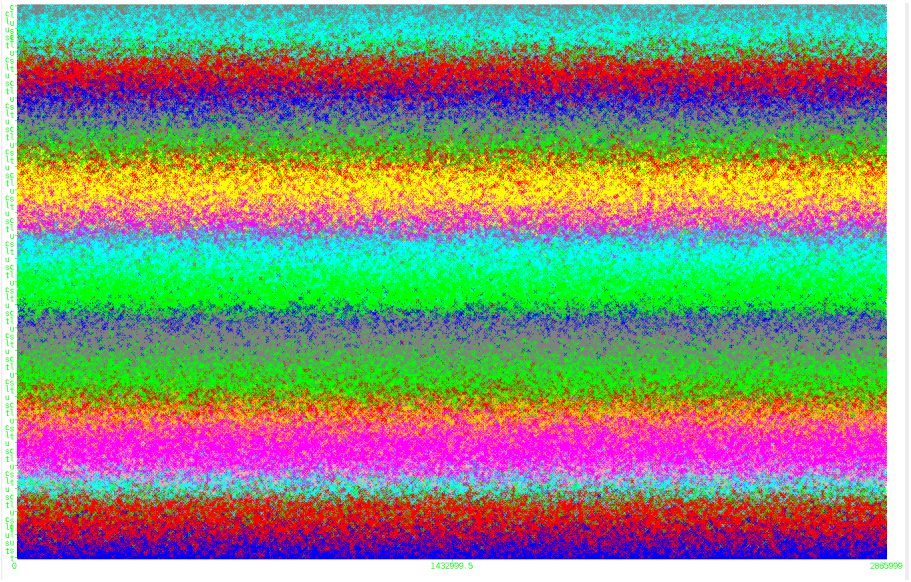

Figure 3.6: Weka K-means Clustering Visualization

representation, the readability of the outputs are not helpful for the big amount of data, which can be seen in the figure 3.6.

3.3 Unsupervised Learning Algorithms Employed

Unsupervised learning is a machine learning approach being employed on unlabeled samples to define and cluster data structure. Self Organizing Maps and K-Means are two well-known unsupervised machine learning algorithms that are employed in this thesis. Note that Weka v3.6.13 [39] and Matlab R2016a are used to implement and run the aforementioned algorithms.

3.3.1 Self-Organizing Map

Self-Organizing Map (SOM) is an unsupervised learning approach introduced in [21] and is a type of Artificial Neural Network that is used to cluster input vectors based on the input space by transforming the high-dimensional data to two dimensional topographical maps. The major advantage of the SOM is that it is easy to interpret and observe the input data even if it is large and complex since SOM provides dimensionality reduction and grid clustering to analyze the similarities of the data. On the other hand, the main drawback of the SOM is that all the neurons must have weights

calculated by employing a large amount of training data to reduce the error rate and make it sufficient to cluster inputs. This approach is beneficial for filtering actions or analyzing the trends of the behaviors in different parts of a network to model input patterns, Eq. 3.1.

$$Wn_{upd} = Wn_{cr} + D(BMU, n) * c(V - Wn_{cr}) \tag{3.1}$$

where Wn_{cr} is the current, Wn_{upd} is the updated weight of the neuron n, V is the input vector and $D(BMU, n)$ represents the Euclidean distance between the BMU and the neuron n.

I employed SOM on MatLab which works based on the Algorithm 1. It should be noted here that SOM assigns weights of nodes randomly at the beginning. Then, during the training phase, the weights are updated based on the characteristics of the data points by the algorithm. At the end of the training phase, the weights converges to the values that represent the data that the SOM is trained on. I employed 30% of the total dataset as training data to increase the consistency of weight values, and the rest of the dataset as the test set.

The SOM presents the data in a map where the neurons having related information are kept closer and both the distribution and the topology of the input data are clear and preserved. In other words, it provides you to cluster the data, but at the same time it orders the clusters.

SOM is a winner-takes-all based algorithm, where the neuron having the more correlated weight, which is calculated by the Euclidean distance of an input vector to the neuron's synaptic weight, is the winner. This is called the Best Matching Unit (BMU) and is able to update its synaptic weight with its neighbour neurons depending on the input vector based on Eq. 3.2.

$$\sigma(t) = \sigma_0 exp(-\frac{t}{\lambda}) \tag{3.2}$$

where σ_0 indicates the width of matching area at time zero, t corresponds the current iteration number and λ is the time constant which depends on σ_0 and the number of iteration for algorithm.

SOM represents the output by N X N neurons on the x-y coordinate system and N stands for the number of neurons for each axis. SOM is used both to reduce

Algorithm 1 Self-Organizing Map Algorithm

Data: Training Data - 30%, Test Data - 70%

Assign weight of nodes via Training Data

while *true* **do**

  Get an input vector V from Test Data;

  $N \leftarrow$ Number of nodes;

  foreach *Node n in N* **do**

    $Distance_{Vn} \leftarrow$ EuclideanDistance(V,n);

  end

  Select the node having the smallest distance as Best Matching Unit BMU;

  Update the weight of BMU and its neighboors based on Eq. 3.1;

  if *iteration limit exceeds* **then**

    break;

  else

    continue;

  end

end

the dimension of data and clustering. Also, SOM uses topological information about which classes are most similar to others to classify data.

Each class is shown by a different color in the map representation and dominant class of a neuron is decided by looking at the color of the neuron. If the color is mixed instead a specific sharp one, this shows the data density for each is balanced. The size of the colored area shows the total amount of data for that neuron.

3.3.2 K-Means

K-means is one of the most commonly used unsupervised learning algorithm that is used to partition a dataset into K groups [17]. The algorithm starts with defining the number of clusters k, and the initial seeds of each cluster. After that, each data instance Xi is assigned to the closest cluster with respect to the result of Euclidean Distance function. It should be noted here that Euclidean Distance is the length of the straight line connecting two points. Then, the cluster centers are re-computed as the mean of assigned instances. Also, to assign the instances to the new clusters

regarding to the new cluster centers are repeated until there is no further change in assignment of instances to clusters. The initial cluster centers can either be defined as certain values or be chosen randomly.The K-means algorithm which works based on the Algorithm 2 is employed in this study.

Let X be the set of instances and x_i be the i th instance of the set. The dataset X is partitioned into the subsets $C_1, C_2, ...C_k$ and j is the mean of the cluster j. The squared error between μ_j and the points in cluster C_j is defined as, Eq. 3.3:

$$SE(Cj) = \sum_{i=1}^{N_j} |x_i - \mu_j|^2 \tag{3.3}$$

The main purpose of the algorithm is to minimize the average distances between instances and the cluster centers. In this regard, K-means algorithm groups the dataset with minimizing the squared error between the mean of a cluster and the points in the cluster. The sum of squared error for all clusters is, Eq. 3.4:

$$SSE(C) = \sum_{i=1}^{N} \sum_{j=1}^{k} ||x_i - \mu_j||^2 \tag{3.4}$$

Determining the number of clusters k is the main issue of K-means algorithm that is depending on the expected partition resolution of the user and characteristics of dataset. Increasing cluster number will reduce the error and the case that number of cluster is equal to the number of instance, has zero error. As a matter of fact, the case does not serve the purpose of the algorithm. The other way round, decreasing the cluster size will cause higher squared error. Thus, the optimal cluster number is located in the range of clustering all the instances into only one cluster and considering each data point as its own cluster. All in all, either one of the suggested approaches from the literature should be applied to determine the optimal number of clusters or the decision should be made by using a priori knowledge of the features of the data set. Note that, the decision of k is assigned to a custom value for this study in order to make more reliable comparison with another clustering method.

3.4 Feature Selection

Feature selection is a data pre-processing step technique for eliminating redundant and less informative features to both decrease the memory usage and computational

Algorithm 2 K-Means Algorithm

Data: Define k, the number of cluster

while *true* **do**

  $k \leftarrow$ Number of nodes;

  Let $C_1...C_k$ be the initial cluster center

  Let $X_1, X_2...X_N$ be the instances

  foreach *Cluster j in k* **do**

    $DistanceV_i \leftarrow$ EuclideanDistance(C_j,X_i);

  end

  Select the cluster having the smallest distance with C_j;

  Update cluster centers;

  if *iteration limit exceeds OR there is no change on cluster centers* **then**

    break;

  else

    continue;

  end

end

cost besides increasing the consistency in the learning algorithms. It is challenging to reveal the most informative features on a dataset since an in-depth analysis would be required. To achieve this, I analyzed all the features in my dataset and defined the following five features as the most informative ones based on the datasets used in this thesis.

3.4.1 Latency

Latency is one of the key terms affecting the performance of a network. It refers to the delay from making a request by a client and performing a response by a server. Reducing the travel time of data, using cut-through switching - which starts transmission of the packet to the destination immediately when its first part arrives rather than waiting for the full packet arrival - and using smaller packets decrease the latency for network systems [2]. Note that it is not possible to decrease the latency under d/V_{light}, where d refers to distance and V_{light} shows the speed of light. The researchers in [20] and [13] explained the importance of the latency on network

optimization in depth.

3.4.2 Hop Count

Hop is a term representing the each network device such as routers between source and destination. When a packet passes through a hop, the hop sends it along to the next hop after processing it. Hop count refers to the number of total hops in the path of a packet from its source to its destination. The researchers in [29] and [14] expressed the relation between the hop count and latency and emphasized that minimizing the hop count increases network performance by decreasing the latency. It should be noted here that calculating the hop count and minimizing it is not difficult when the topology of a network is known.

3.4.3 Probe and Reply TTLs

Time-to-live (TTL) is a metric each network packet has showing that how many hops the packet can travel at maximum. A packet is forwarded toward its destination and its TTL value is decreased by 1 after each step. Therefore, the packet should reach its destination before its TTL value becomes zero, otherwise it would be dropped. This helps limiting to flood a network with a request.

TTL also plays a major role in increasing caching performance by eliminating misrouting overhead caused by route errors. Assigning large TTL values helps the packets traveling over the network longer and reach its destination before being discarded. However, it might cause the usage of invalid routes that results in extra routing if the TTL is too high. On the other hand, keeping small TTL values causes the packets getting discarded before they reach to their destination. Researches in [30] emphasized that TTL should be much lesser than commonly seen in the networks to save bandwidth. Authors in [24] presented different methods to define the optimal TTL values to minimize routing delay. Note that TTL value of a packet can be assigned between 1 and 255.

To the best of my knowledge, this is the first work focusing on clustering by SOM and the effects aforementioned features to optimize the service / server deployment process. In order to analyze this further, I employ publicly available datasets collected in nine different countries over three different continents by CAIDA. I also employ

Neural Network Toolbox and Weka system, to run the SOM and the K-means algorithms, respectively, on an dataset with their default configurations to reveal the most relevant records to eliminate the noise. Using publicly available datasets and tools ensures that my research can be easily validated and compared against others in the field.

3.5 Performance Metrics

While aiming to improve the performance of whole system, it is indispensable not to mention about the performance of the proposed method. Thus, the performance is discussed especially for the following steps of the study;

- Data pre-processing
- Feature Selection method
- Determining clustering parameters
- Evaluation of the results

First of all, the dataset should be converted into required formats to be analyzed by Matlab and Weka. As is mentioned before, even though Matlab is able to run both SOM and K-means algorithms, Weka is also used to provide an open-source alternative for this thesis. In that case, I need to handle the efficiency of converting data into different formats more than once. In order to use Matlab, the data should be converted to .dat format which is basically row-column representation of data attributes and data instances. Weka has a specific input data format, called as arff. CAIDA collects the dataset in wartz format and the provided API is able to convert dataset into only pcap, json, text, cat and dump formats, directly. Data is converted to JSON because Java has Json libraries that makes reading, editing and saving data more efficient. A Java application is developed for the data conversion and also makes feature selection at the same time. The performance of this application is measured for all dataset. It is measured that the application handles 1000 data instances in 14 sec on average. The application not only selects the attributes but also makes some calculations about the network metrics.

In order to analyze the performance more deeply, the complexity of the algorithm can be discussed. The algorithm is written in $O(n)$ complexity and n indicates the total number of instances, except the analysis part of each hop. The complexity of the analysis of the hop count part is O(n X *averagehopcount*). The code has the lowest possible complexity and does not reduce the performance of whole system.

The second Java application is developed to convert data into arff format by using the libraries that Weka provides. This application gets the input file that the first java application generates. However, the feature selection is already done and the second application just converts data into arff format, this application is not faster than the first one. This means the first application has the same complexity as Weka libraries.

Determining the number of cluster is the common problem for all clustering applications. In this study, I applied the algorithm using different cluster numbers and check both the performance and the ability of interpret the results of method. First of all, the dimension size of SOM algorithm is defined as 10 X 10. As in the figure 3.7, the dispersion of the instances into clusters makes harder to emphasize the candidate clusters. Since, there are more number of clusters as to be considered as candidate. Both, the performance of generation of the SOM results by Matlab and the analysis of the cluster results will be affected by the dimension size so the dimension size is reduced to 5 x 5. The dimension size lower than 5 X 5 is not selected because of the higher mean sum of squares (MSS) error of clustering. With this regard, the cluster size is defined as 25 for K-means algorithm, as well.

In order to interpret the cluster result to decide the optimal location, it is required to find the average scores for all attributes for all candidate clusters. Weka has already provided the average scores and it is not necessary to calculate the scores. However, Weka does not provide the information of the number of data instances for the clusters, by default. So I coded another Java application to find the data amounts by clusters of Weka results. Moreover, Matlab does not provide the average attribute scores. Thus, I coded a Matlab script in order to find cluster statistics, instead of using Java. That is why, I could compare the performance of the two methods. Matlab provides better performance to make the calculations on the whole dataset.

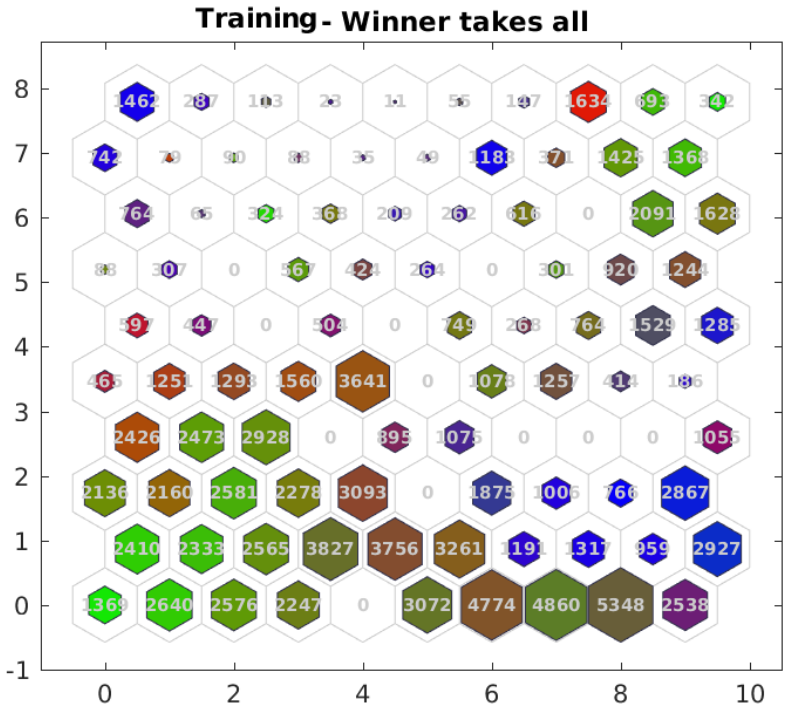

Figure 3.7: Example Matlab Output Result with 10x10 Dimension Size

The SOM algorithm generates the clustering results in 7-10 minutes, however the K-means algorithm finishes in 4-6 minutes minutes on the average. As is mentioned in the paper [31], the K-means algorithm is faster than the SOM algorithm with the same datasets and cluster sizes.

Chapter 4

Experiments and Results

This chapter focuses on the analysis, experiments and evaluations that were carried out during the research. At a glance, a novel approach to find out the most optimal location(s) to place server(s) to manage a large network with the maximum efficiency by using Self-Organizing Map and K-means Algorithm. Additionally, the advantages and disadvantages of the proposed approach are presented.

4.1 Revealing the Characteristics of Network Performance Metrics and Optimization Method

My aim in this section is to find the most appropriate answers for the following questions:

1. What are the network properties that should be employed to define the optimal location of a service on the cloud? How should those properties be used?

2. Which unsupervised algorithm provides the most optimal results in terms of optimization and how should they be applied?

3. What are the advantages and disadvantages of the unsupervised methods such as SOM and K-means that are planning to be used?

To this end, I focused on nine publicly available forward-IP path datasets collected over three different aforementioned regions (see Chapter-3) from all over the world for the years 2012, 2013 and 2014. The different years are selected for this experiment to find out how and why the optimal location(s) change over different years (time periods). This research also gives an insight into the features that are more informative to use for building a machine learning based clustering approach to analyze cloud network data.

4.1.1 Development of Java Application

Before applying the unsupervised algorithms, some preprocessing operations should be applied on the employed datasets such as feature selection, eliminating unused features, data discretization and re-formatting data. In order to apply these preprocessing steps, a Java application is developed using Json libraries. Each packet / instance is considered as a Json object and separated into attributes. Note that the application does not only apply the preporcessing steps, but also finds the total and the average scores for the desired attributes as shown in the algorithm 3. Lastly, the formatted instances are written into the new output file. The reason for reading the instances from an input file and calculating the sum and average scores at the same time is to reduce the complexity of the algorithm. As is seen in the algorithm, the complexity is $O(Number\_of\_instances$ x $Average\_hop\_count)$ which is the least possible complexity to handle all instances and hop counts.

Algorithm 3 Java Preprocessing Application Algorithm

Data: Dataset File

Create the file reader *br*

Create a list *dataList* s for all instances with the type *Data*

Create *data* object with the type *Data*

while *dataInst read line from br is not Null* **do**

    Parse the Json object in *attr*

    Initialize each *class* using source IP

    Label $failure$ values

    while *hop is not Null* **do**

        Count *hop* s

        Sum *ttl* values

        Sum *rtt*

    end

    Set all attributes of *data* object

    Add *data* object into *dataList*

    Write into new output file with the new format

end

Find average attribute scores

4.1.2 Data Clustering

Clustering is the division of data into groups with respect to similarities. In other words, the instances that are grouped into the same cluster are more similar to each other and less similar to the instances from other clusters [3]. In this research, clustering algorithms are applied for the selected datasets in order to decide the most optimal location of a service on the cloud in terms of the features explained in Chapter 3.4. This research is aimed to support the human expert (network manager) for setting up new servers, or a main server, to the most effective location from a given set of locations.

To perform the experiments on this purpose, three main regions, namely Asia Pacific, North America and Europe are selected; where the locations Australia, Japan and South Korea are selected from Asia Pacific, San Jose, Virginia and Ottawa are chosen from North America, and Germany, Ireland and England are selected from Europe. Those regions are the ones including the most number of monitors according to CAIDA [5], and the cities selected are located far from each other to observe the effect of hop count and RTT in detail. For each region, the SOM and K-means algorithms are applied for three different years, 2012, 2013 and 2014. The results for each region can be used for making the decision of the optimal location given a region. Also, the results for each region can be compared and the decision can be made among the main locations.

SOM

SOM is one of the most commonly used neural network algorithm which is inspired by human brain. SOM reduces the dimension of input space and produces low-dimensional, typically two-dimensional, output. In order to use SOM for clustering, the objects in the same node in the input space are regarded as grouped in the same cluster. The most important advantage of SOM is low-dimensional representation which makes the results easy to interpret. Two-level approach is applied for this experiment. The first level refers finding the optimal location for each main region and all inter-region results are compared as overall optimal location for the second level. The reason of using two level is to compare only the inter-region results instead of comparing all sub-locations.

Each node will be identified by sequential numbers that are increased from left to right and from top to bottom. The top-left node has the node id 1. The nodes might also be called as cluster or group afterwards. The primary colors blue, red and green represent the classes which corresponds to locations in my case. If a node has a pure color, it means there is a majority class for the node. Mixed colored nodes have more homogeneous data. The numbers on each node represent the total number of data that is grouped into that cluster. In this section, the SOM results for each year and location is explained.

2012

The Figure 4.1 shows SOM results for Asia Pacific 2012 dataset in 5x5 map. Each location is considered as a different class. South Korea is represented by red (Class 1), Australia is represented by green (Class 2) and Japan is shown by blue (Class 3). The nodes having the largest amount of data are considered as candidate nodes since those nodes have more similar data, which affect the overall results more than the other nodes. The rule for selecting the candidate clusters is based on the Eq. 4.1 of determining the sample size for a population with a selected confidence level in the case of population size is known [22].

$$Min.ClusterSize(S) = \frac{X^2 * N * P * (1 - P)}{d^2 * (N - 1) + X^2 * P * (1 - P)} \tag{4.1}$$

where X^2 represents the table value of Chi-Square for desired confidence level, N refers to the population size, P shows the population proportion and d refers to the degree of accuracy. The relationship between sample size and population size is illustrated in the paper [22] which shows that as the population size increases, the sample size increases at a decreasing rate and remains relatively constant around a certain sample size. For instance, both the population sizes 1000000 and 2000 require a similar number of sample size. Thus, the usage of this equation may cause high cost and performance issues since it returns a small value for S regardless of the confidence level used in calculating X^2, which causes one to select more nodes as candidates. Therefore, Eq. 4.2 is generated to provide a consistent value for S

$$Min.ClusterSize(S) = (T_{size}/D_{size}) * coef \tag{4.2}$$

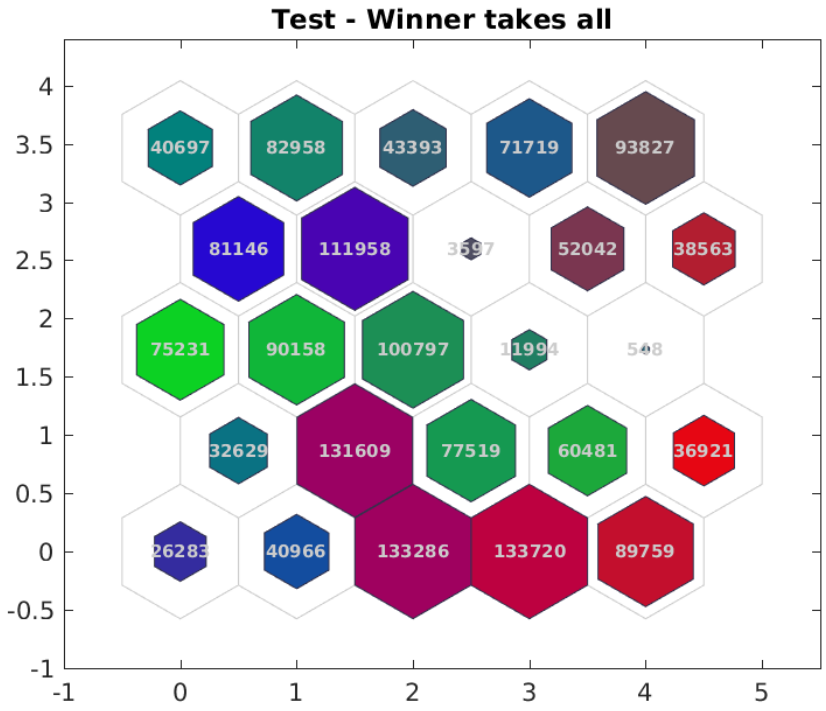

Figure 4.1: Asia Pacific 2012 SOM Result

where T_{size} refers to the total training data size. D_{size} represents the dimension size (5x5), $coef$ is a coefficient that should meet the requirement of the minimum sample size and also should suggest an affordable number of candidate clusters to be analyzed without negatively affecting the performance. For that reason, it is defined as 1.5 for all experiments. Note that the number of candidate clusters to be selected changes based on Eq. 4.2, which focuses on the data dispersion and clustering results.

For Asia Pacific 2012 datasets, 5 nodes / clusters, 7th, 13th, 17th, 23rd and 24th, are the major ones having the largest colored area because they have the most number of similar data grouped into those nodes. Also, those nodes meet the minimum S requirement by Eq. 4.2 and are selected as candidate clusters.

The amount of the objects that are grouped into the clusters are shown in Figure 4.1. It is seen that:

- 7th node contains 31848 South Korea data (red), 1846 Australia data (green) and 78264 Japan data (blue).
- 13th node contains 10885 South Korea data (red), 56408 Australia data (green)

and 33504 Japan data (blue).

- 17th node contains 79751 South Korea data (red), 685 Australia data (green) and 51173 Japan data (blue).
- 23rd node contains 83231 South Korea data (red), 551 Australia data (green) and 49504 Japan data (blue).
- 24th node contains 99335 South Korea data (red), 573 Australia data (green) and 33812 Japan data (blue).

The table 4.1 shows the number of data for each clusters by classes. Observing the amounts for each class helps when the color of the cluster is not certain, occurred by the combination of at least two colors, and the ratio of mixed colors are close to each other. In that case, the majority class can not be understood by manually looking at the color of the cluster. For instance, it is straightforward to define the majority class when the color is close to pure red, blue or green, but when the color is purple like the 23rd node, it is difficult to decide which class is included more than other. Since it is known that purple is a mix of red and blue, it is concluded that the majority classes for the cluster are Class 1 and Class 3, but this node contains Class 2 data as well.

The Figure 4.2 shows SOM results for Europe 2012 dataset in 5x5 map. Ireland is represented by red (Class 1), Germany is represented by green (Class 2) and the United Kingdom is represented by blue (Class 3). The 7th, 13th, 18th and 24th nodes have the largest colored area by the reason of having the most number of similar data grouped into those nodes. Those nodes are candidate of being selected as optimal cluster because of having desired data amount.

The amounts of the objects that are grouped into the clusters N7, N13, N18 and , N24 are 181177, 162313, 167499 and 179480, respectively. The table 4.2 shows the number of data for only the candidate clusters by classes as:

- N7 contains 5187 Ireland data (red), 55686 Germany data (green) and 120304 UK data (blue).
- N24 contains 56387 Ireland data (red), 67659 Germany data (green) and 55434 UK data (blue).

Table 4.1: Data Amounts for Each Node For Asia Pacific 2012 dataset

Node	**Class 1**	**Class 2**	**Class 3**	**Total**
N1	256	20598	19843	40697
N2	5811	42760	34387	82958
N3	7845	15967	19581	43393
N4	8105	24783	38831	71719
N5	37611	27124	29092	93827
N6	12207	2498	66441	81146
N7	**31848**	**1846**	**78264**	**111958**
N8	1108	1105	1384	3597
N9	24842	10964	16236	52042
N10	26844	4497	7222	38563
N11	3402	61571	10258	75231
N12	5666	64243	20249	90158
N13	**10885**	**56408**	**33504**	**100797**
N14	1490	5903	4601	11994
N15	85	212	251	548
N16	1250	14554	16825	32629
N17	**79751**	**685**	**51173**	**131609**
N18	6254	46556	24709	77519
N19	6757	39804	13920	60481
N20	33364	851	2706	36921
N21	5363	4504	16416	26283
N22	3062	12362	25542	40966
N23	**83231**	**551**	**49504**	**133286**
N24	**99335**	**573**	**33812**	**133720**
N25	68647	5182	15930	89759

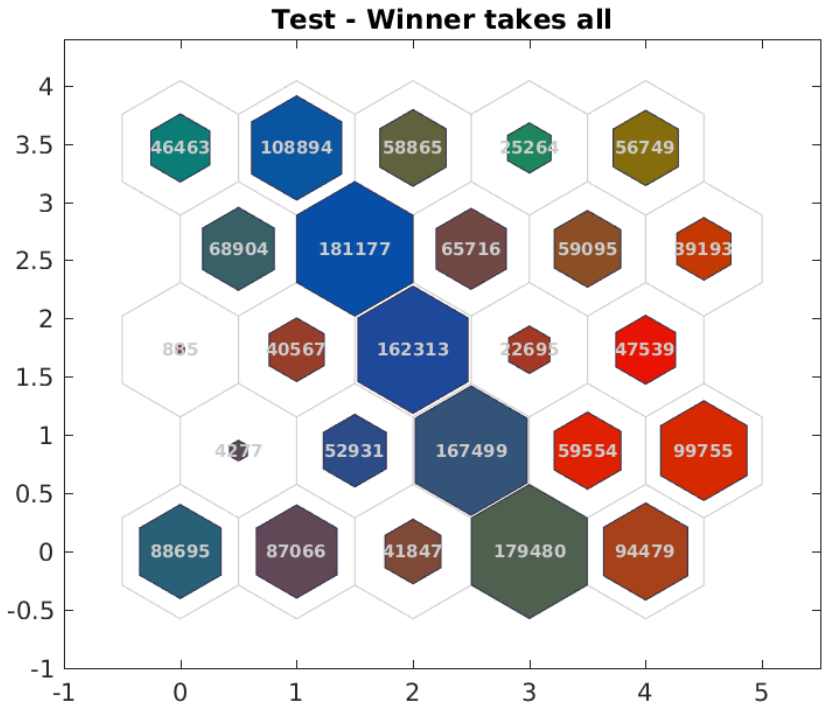

Figure 4.2: Europe 2012 SOM Result

Table 4.2: Data Amounts for Candidate Nodes For Europe 2012 dataset

	Class1	**Class2**	**Class3**	**Total**
N7	5187	55686	120304	181177
N13	19205	45655	97453	162313
N18	34318	54234	78947	167499
N24	56387	67659	55434	179480

- N18 contains 34318 Ireland data (red), 54234 Germany data (green) and 78947 UK data (blue).

- N13 contains 19205 Ireland data (red), 45655 Germany data (green) and 97453 UK data (blue).

The Figure 4.3 shows SOM results for North America 2012 dataset in 5x5 map. Virginia is represented by red (Class 1), Ottawa is represented by green (Class 2) and San Jose is represented by blue (Class 3). N4, N8, N13, N24 and N25 have the largest amount of data satisfying the minimum cluster size. As is shown in the SOM result, N4, N8 and N25 has purer colors which are close to green and blue. For those

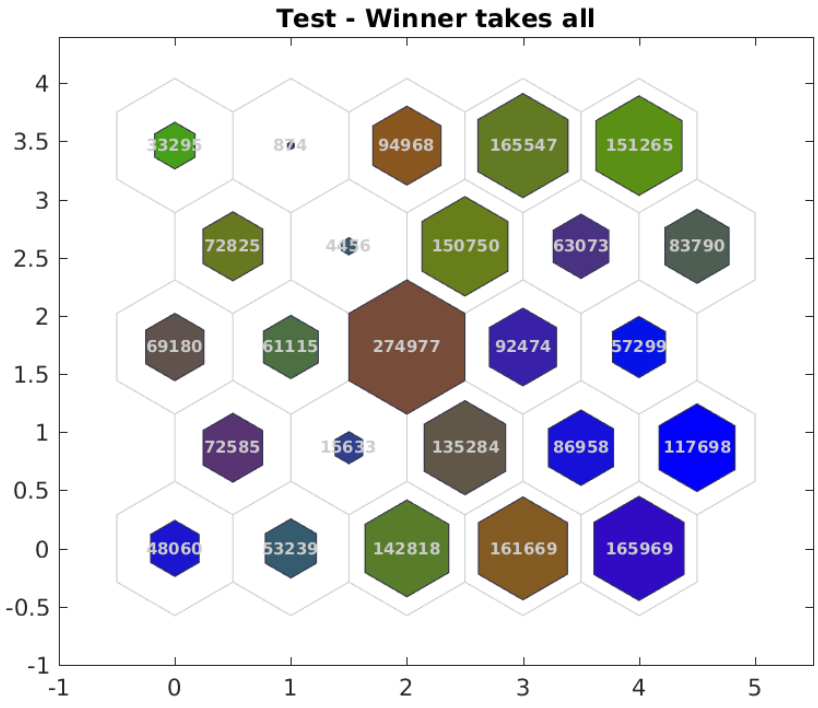

Figure 4.3: North America 2012 SOM Result

Table 4.3: Data Amounts for Candidate Nodes For North America 2012 dataset

	Class1	**Class2**	**Class3**	**Total**
N4	63366	79710	22471	165547
N8	61126	74476	15148	150750
N13	128467	83464	63046	274977
N24	83130	57003	21536	161669
N25	31752	7371	126846	165969

nodes defining the majority class is easier. However some nodes have mixed colors because of having similar amount of data instances for each class. For instance, N24 is represented by brown which shows most of the data instances belong to red (Virginia) and green (Ottawa).

The amounts of the objects that are grouped into the clusters N4, N8, N13, N25 and N24 are 165547, 150750, 274977, 165969 and 161669, respectively. The table 4.3 shows the number of data for candidate clusters by classes in detail.

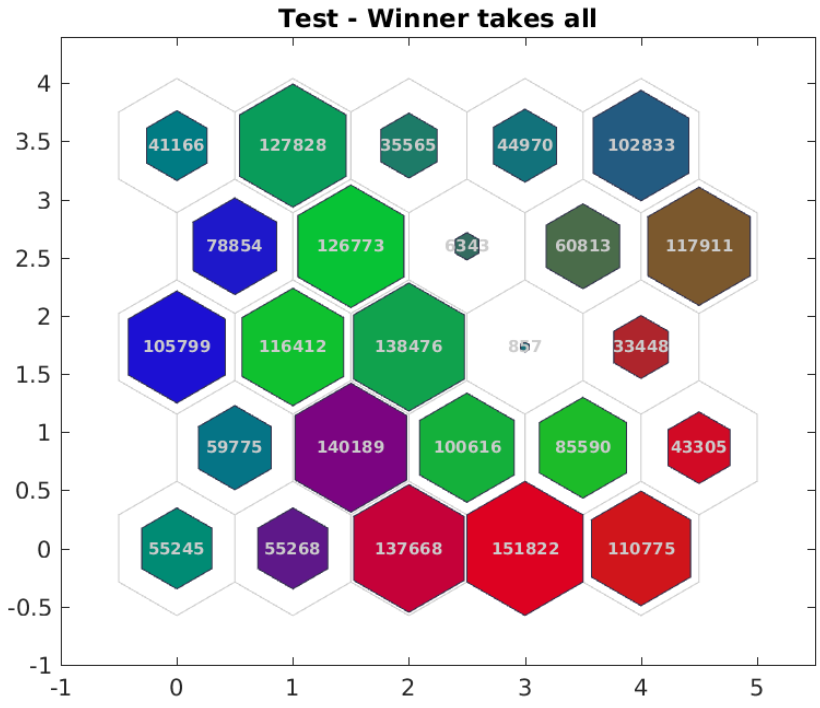

Figure 4.4: Asia Pacific 2013 SOM Result

2013

SOM results for Asia Pacific 2013 dataset is shown in the Figure 4.4. Each location is considered as a different class. As in the 2012 dataset, South Korea is represented by red (Class 1), Australia is represented by green (Class 2) and Japan is represented by blue (Class 3). N2, N7, N13, N17, N23 and N24, which have the largest amount of data, are selected as candidate clusters. As to be seen, the color of the nodes are closer to the main colors that shows there is a majority class for all candidate clusters.

The total amount of the objects that are grouped into the clusters and the object size by classes are represented in the table 4.4.

The Figure 4.5 represents the SOM results for Europe 2013 dataset in 5x5 map. Ireland is represented by red (Class 1), Germany is represented by green (Class 2) and the United Kingdom is represented by blue (Class 3). The 7th, 13rd, 18th, 24th and 25th nodes are selected as candidate clusters. Obviously, only N7 has a color which help to understand the majority class. However, it is hard to interpret the majority class for the other clusters.

Table 4.4: Data Amounts for Candidate Nodes For Asia Pacific 2013 dataset

	Class1	Class2	Class3	Total
N2	4747	78134	44947	127878
N7	3616	96980	26177	126773
N13	8872	88054	41550	138476
N17	67528	1969	70692	140189
N23	106353	433	30882	137668
N24	131633	8930	11564	110775

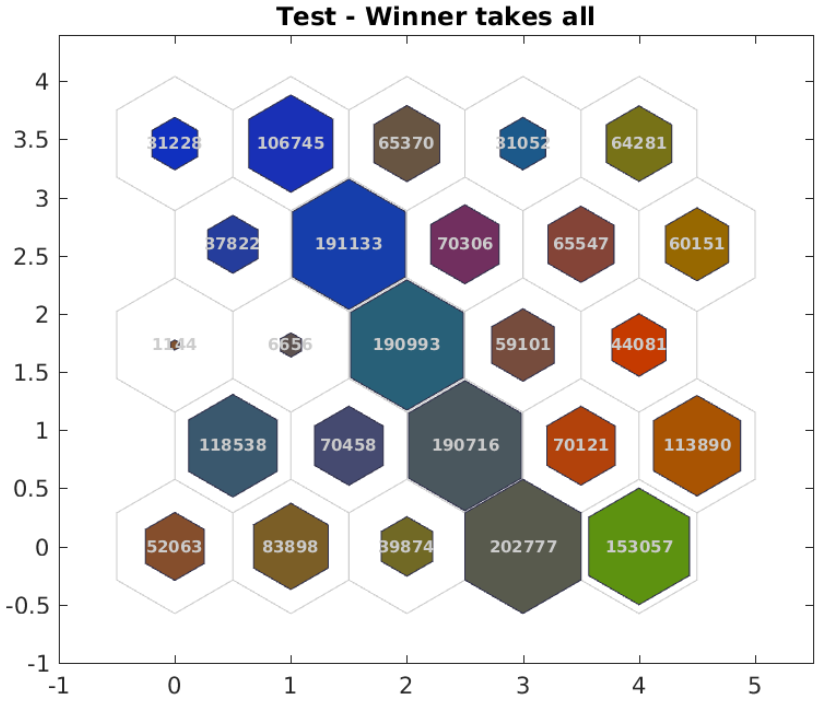

Figure 4.5: Europe 2013 SOM Result

Table 4.5: Data Amounts for Candidate Nodes For Europe 2013 dataset

	Class1	**Class2**	**Class3**	**Total**
N7	16681	46190	128262	191133
N13	29840	71567	89586	190993
N18	55335	64712	70669	190716
N24	70223	71636	60918	202777
N25	55652	87531	9874	153057

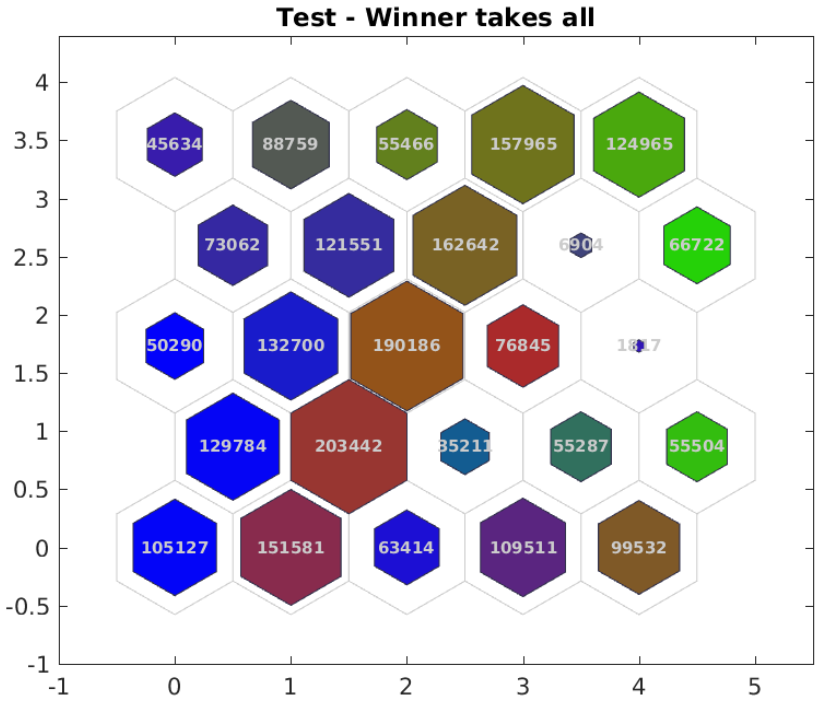

Figure 4.6: North America 2013 SOM Result

The amount of the objects that are grouped into the candidate clusters can be seen in the table 4.5 . As the table shows, there are at least two classes that have close number of data size in the nodes except N7.

The Figure 4.6 shows the SOM results for North America, 2013 dataset. Virginia is represented by red (Class 1), Ottawa is represented by green (Class 2) and San Jose is represented by blue (Class 3). The nodes N4, N8, N13, N17 and N22 satisfies the minimum number of cluster size and they are selected as candidate clusters.

The total amount of the objects that are grouped into the candidate clusters and the number of data included by classes are represented in the table 4.6.

Table 4.6: Data Amounts for Candidate Nodes For North America 2013 dataset

	Class1	Class2	Class3	Total
N4	69035	71308	17622	157965
N8	77045	62544	23053	162642
N13	109921	61581	18684	190186
N17	120877	43330	39235	203442
N22	80790	25277	45514	151581

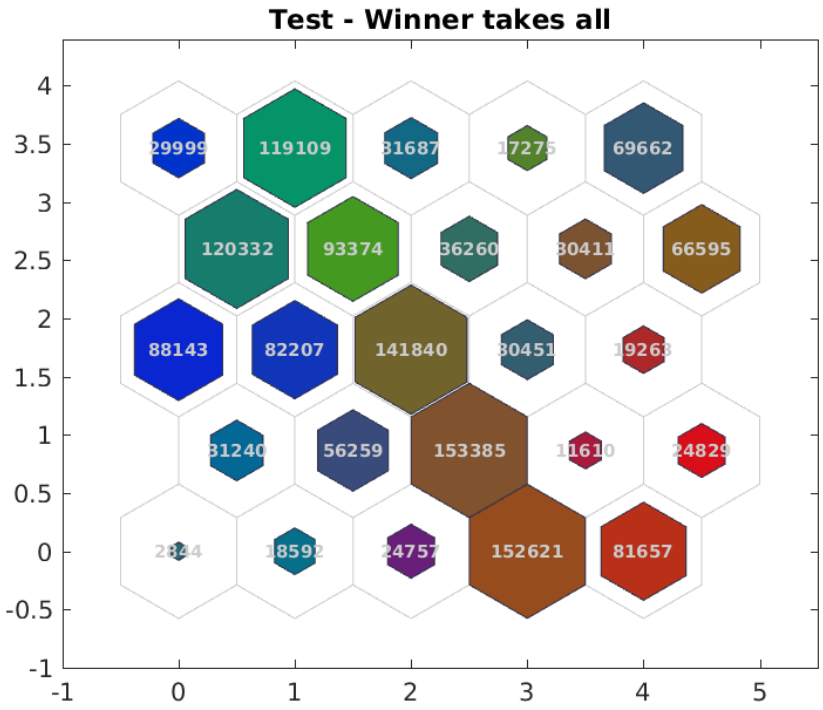

Figure 4.7: Asia Pacific 2014 SOM Result

2014

The Figure 4.7 provides the SOM results for Asia Pacific 2014 dataset in 5x5 map. South Korea is represented by red (Class 1), Australia is represented by green (Class 2) and Japan is represented by blue (Class 3). The candidate nodes N2, N6, N7, N13, N18 and N24 satisfies the minimum cluster size and are selected to analyze in depth.

The exact data amounts for each class located into each candidate cluster and total data amount for each cluster are shown in the Table 4.7.

The Figure 4.8 presents the SOM result for Europe 2014 dataset in 5x5 map. Ireland is represented by red (Class 1), Germany is represented by green (Class 2)

Table 4.7: Data Amounts for Candidate Nodes For Asia Pacific 2014 dataset

	Class1	**Class2**	**Class3**	**Total**
N2	1646	69236	48227	119109
N6	10394	58748	51190	120332
N7	24822	56390	12162	93374
N13	62431	55158	24251	141840
N18	76850	49232	27303	153385
N24	89794	45450	17377	152621

Table 4.8: Data Amounts for Candidate Nodes For Europe 2014 dataset

	Class1	**Class2**	**Class3**	**Total**
N4	11627	34818	100555	147000
N5	5863	22911	69744	98518
N8	16115	36451	65883	118449
N13	39417	57776	56102	153295
N17	49320	55926	36872	142118
N21	74231	52628	3007	129866
N22	43200	52851	2450	98501

and the United Kingdom is represented by blue (Class 3). N4, N5, N8, N13, N17, N21 and N22 are selected as candidate clusters. There are more number of candidate clusters here, that means the data is more distributed in comparison with other data sets. The table 4.8 shows the number of data for candidate clusters by each classes.

The Figure 4.9 shows SOM result for North America 2014 dataset in 5x5 map. Virginia is represented by red (Class 1), Ottawa is represented by green (Class 2) and San Jose is represented by blue (Class 3). N6, N7, N8, N9, N11 and N12 are selected as candidate clusters.

The table 4.9 presents the data amounts for each class and the total data amounts located into each candidate clusters.

K-means

K means algorithm is applied for three different locations and three different years by employing Weka. Note that each dataset contains millions of records, therefore the memory space of Weka is increased in its configuration file to 2048 MB from 512 MB.

To ensure the comparison of SOM and K-means algorithms is reliable, the same properties, such as dimension, distance algorithm, number of attributes are used.

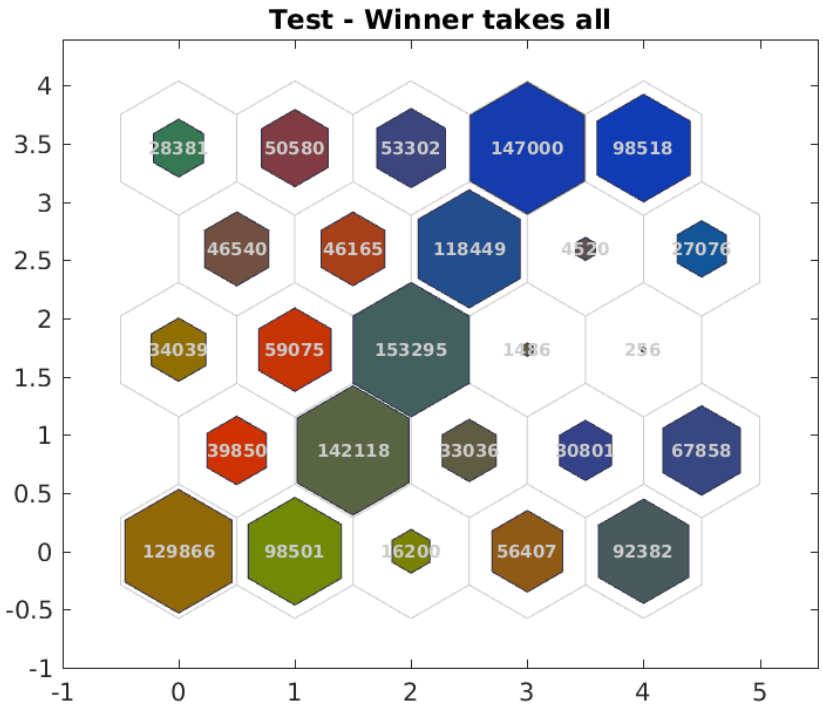

Figure 4.8: Europe 2014 SOM Result

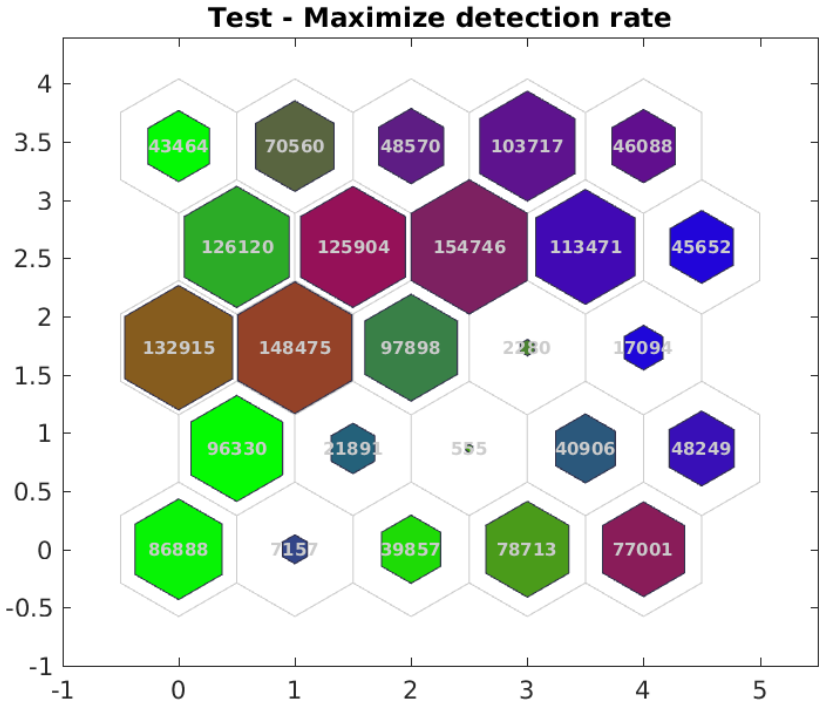

Figure 4.9: North America 2014 SOM Result

Table 4.9: Data Amounts for Candidate Nodes For North America 2014 dataset

	Class1	**Class2**	**Class3**	**Total**
N6	20327	91028	14765	126120
N7	77817	10524	37563	125904
N8	79571	24508	50667	154746
N9	33172	5547	74752	113471
N11	67178	53620	12117	132915
N12	85377	44155	18943	148475

Euclidean method is selected as distance function for 25 clusters and the default seed, which is 10, is used. The dataset is clustered with respect to five attributes, hop count, latency (RTT), probe TTL, reply TTL and failure.

The most explicit drawback of K-means is the lack of having low dimensional mapping representation in comparison with SOM. The data points can be shown in coordinate system by selecting any attribute for x and y axis and the instance number-clusters can also be selected in order to see which data instance is grouped into which cluster, but the representation is difficult to interpret unlike SOM. That is why only summary tables are provided by K-means algorithm instead of showing graphical representation.

The default output of Weka for K-means algorithm contains the centroid information for each attribute and the total amount of data for each cluster. A Java application is developed to obtain the data amounts by classes for all clusters. The summary table 4.10 shows the candidate clusters for the output of the K-means algorithm for the years 2012, 2013 and 2014. The number of candidate clusters for Asia Pacific dataset are 7, 4 and 6 for the years 2012, 2013 and 2014, respectively. 4, 5 and 4 candidate clusters are selected for the Europe datasets for the years 2012, 2013 and 2014, respectively. In the same manner, the number of candidate clusters for the North America datasets are 6, 8 and 5 for each selected years in a row.

Needless to say, there are less candidate clusters for Europe dataset than the other locations in 2012 which shows the data is less dispersed into different clusters and more likely, there are a greater number of similar instances in this dataset. Note that the corresponding locations for each class are same as the SOM clustering for all datasets in all selected years.

Table 4.10: Data Amounts for Candidate Nodes for all datasets in 2012 for K-means algorithm

Year	Location	Cluster No	Class1	Class2	Class3	Total
2012	Asia P.	1	12120	79692	65219	157031
		3	64292	38881	53998	157171
		11	88748	28400	38822	155970
		18	49125	43176	56244	148545
		20	50590	87878	126499	264967
		21	65039	27810	55998	148847
		23	90152	76681	100429	267262
	Europe	2	97464	85431	112503	295398
		4	28130	72319	134741	235190
		6	104765	93907	75758	274430
		10	14226	73162	146946	234334
	N. America	2	74736	74670	87742	237148
		6	114233	141892	28485	284610
		7	71416	89344	64731	225491
		14	101820	52355	67716	221891
		15	163775	118963	18938	301676
		20	91945	46527	85643	224115
2013	Asia P.	2	56371	65813	76261	198445
		5	30077	76470	78905	185452
		18	15025	116280	73862	205167
		21	114726	132847	125654	373227
	Europe	2	80906	55423	48040	184369
		11	48614	52799	93207	194620
		20	86048	96386	158858	341292
		22	81563	72704	29268	183535
		24	91970	93854	201918	387742
	N. America	5	49866	20378	142873	213117
		6	120050	67308	73978	261336
		8	97196	68843	46297	212336
		9	114173	78472	95682	288327
		13	135951	123206	47711	306868
		14	92669	49801	115193	257663
		16	86090	145358	18126	249574
		23	172507	72211	128132	372850
2014	Asia P.	4	2290	93251	76548	172089
		10	28477	135847	155139	319463
		17	68532	105023	114839	288394
		18	58913	51503	41962	152378
		21	47934	53777	50131	151842
		23	64859	53375	29951	148185
	Europe	3	39779	43245	100946	183970
		4	47270	72300	71589	191159
		20	89708	88196	32895	210799
		22	135560	104229	4448	244237
	N. America	1	62807	15231	117966	196004
		7	52557	27362	98458	178377
		14	55362	93406	11973	160741
		16	124781	97437	50742	272960
		20	157268	80490	148305	386063

4.1.3 Decision Mechanism

In this section, my main objective is to reveal how to use cluster results to select the optimal cluster(s). The problem is choosing the optimal cluster(s) out of the candidate clusters considering all the attributes at the same time. The problem of multiple criteria decision making is concerned with mathematical optimization problems involving more than one objective function to be optimized simultaneously. In this regard, I analyzed the dataset and the cluster results to define generalized rules for selection.

In the clustering algorithms, each point is assigned to one and only one centroid and the points assigned to the same centroid belong to the same cluster.Each centroid is the average of all the points belonging to its cluster, so centroids can be treated as data points in the same node. The average scores of attributes for candidate clusters in 2012 is shown in the table 4.11. The average scores of a cluster will be taken into consideration on behalf of all the data scores grouped into the cluster.

Cluster Results

In order to decide the most optimal location within the candidate clusters, the average scores of all attributes for all clusters need to be analyzed. With respect to the attribute results, the maximization and minimization decisions should be made for all attributes.

Table 4.11 shows the average attribute results for the SOM clustering algorithm. Matlab provides a $dimension\_size\ X\ number\_of\_instance$ matrix which contains the cluster information for all belonging instances. A java application was developed to find the average scores using the matrix.

In the same manner, the attribute results for the K-means algorithm are shown in table 4.12. Unlike Matlab, Weka provides average attribute results for each cluster. However, the results are not specific for classes. While making a decision of optimal location, having results by classes is required, since the classes stand for different locations.

Table 4.11: Attribute results for SOM Clustering

Year	Location	Cluster No	Hop Count	RTT	Probe TTL	Reply TTL
2012	Asia P.	7	14.549	18.872	13.006	162.344
		13	16.603	62.249	14.003	170.772
		17	16.582	20.713	14.004	173.725
		23	18.475	28.249	14.983	181.319
		24	20.944	37.111	15.986	189.476
	Europe	7	20.953	20.010	15.990	185.699
		13	18.515	16.641	14.990	179.053
		18	16.440	11.871	14.029	171.870
		24	14.689	11.742	13.014	163.281
	N. America	4	18.625	32.465	14.972	171.649
		8	16.513	15.537	13.997	169.034
		13	14.538	20.086	13.008	157.445
		24	10.627	7.007	9.007	129.372
		25	9.784	12.106	9.036	95.510
2013	Asia P.	2	10.448	26.050	9.042	123.782
		7	12.580	43.792	12.008	149.630
		13	16.668	61.279	14.009	168.557
		17	16.505	18.156	13.870	166.664
		23	18.714	23.611	14.976	176.637
		24	21.011	31.709	16.008	185.778
	Europe	7	20.934	29.947	15.981	185.294
		13	18.534	26.603	15.016	178.866
		18	16.450	22.482	14.036	172.206
		24	14.574	16.375	13.017	161.765
		25	12.668	11.428	12.009	149.828
	N. America	4	18.578	27.369	14.965	171.627
		8	16.561	14.061	13.979	164.747
		13	14.700	16.182	13.007	160.476
		17	12.694	13.758	12.005	145.536
		22	10.359	7.270	9.020	123.658
2014	Asia P.	2	10.784	20.054	9.176	121.991
		6	12.710	29.395	12.005	143.290
		7	14.601	40.473	13.015	163.134
		13	16.724	38.773	14.007	171.395
		18	18.632	41.625	14.971	177.404
		24	21.024	48.769	15.978	183.631
	Europe	4	21.038	27.879	16.007	184.639
		5	24.365	37.679	17.340	188.231
		8	18.383	22.388	15.033	179.524
		13	16.587	21.071	13.997	170.677
		17	14.506	16.752	13.018	161.695
		21	10.038	5.420	9.491	122.961
		22	12.610	11.806	12.013	149.242
	N. America	6	13.327	22.809	12.201	112.760
		7	14.687	15.990	13.009	159.931
		8	16.639	23.927	13.970	163.491
		9	21.288	30.830	16.113	176.877
		11	10.269	8.853	9.043	119.896
		12	12.548	14.610	12.020	142.685

Table 4.12: Cluster results for K-means algorithm

Year	Location	Cluster No	Hop Count	RTT	Probe TTL	Reply TTL
2012	Asia P.	1	11.601	27.730	11.591	140.864
		3	17.000	39.899	13.974	174.610
		11	21.456	51.204	15.955	189.409
		18	13.517	94.422	10.401	229.612
		20	14.472	35.449	12.986	162.714
		21	17.999	40.936	15.002	180.356
		23	15.228	76.054	13.341	223.802
	Europe	2	16.457	20.439	13.951	169.808
		4	19.507	22.244	15.430	181.279
		6	14.672	16.298	13.001	162.793
		10	21.444	30.754	15.944	185.368
	N. America	2	13.011	18.024	12.001	143.563
		6	17.467	33.082	14.384	171.467
		7	11.999	13.231	11.886	137.896
		14	13.999	20.809	12.955	151.872
		15	15.570	26.541	13.554	166.313
		20	10.106	8.131	10.000	115.636
2013	Asia P.	2	17.000	43.244	13.936	166.076
		5	15.000	40.628	12.989	157.473
		18	11.575	31.752	11.564	136.090
		21	15.410	76.252	13.870	214.630
	Europe	2	15.000	18.165	12.989	156.188
		11	19.276	64.373	19.123	203.75
		20	20.456	31.231	15.826	165.416
		22	12.544	48.541	9.023	211.44
		24	23.154	38.949	16.674	166.437
	N. America	5	9.612	8.339	6.999	95.743
		6	16.000	25.575	13.822	148.624
		8	17.000	28.951	13.926	154.281
		9	15.000	21.514	13.926	154.281
		13	18.406	30.049	14.797	159.378
		14	14.000	21.843	12.869	138.169
		16	20.856	32.493	15.816	167.416
		23	12.576	15.508	14.849	11.995
2014	Asia P.	4	9.696	16.837	8.802	110.567
		10	13.132	30.474	12.310	142.221
		17	15.541	37.002	13.437	157.615
		18	18.000	43.589	14.633	167.793
		21	17.000	42.114	13.910	164.183
		23	17.245	73.778	15.062	216.648
	Europe	3	22.439	38.138	16.241	169.033
		4	16.512	21.128	13.985	171.912
		20	13.580	14.337	12.491	149.957
		22	10.573	8.154	9.133	127.611
	N. America	1	10.464	54.034	5.989	186.741
		7	13.000	22.133	11.979	128.815
		14	20.472	31.494	15.652	161.553
		16	17.452	26.544	14.213	154.742
		20	14.477	20.542	12.862	140.373

The effect of Hop Count and RTT on the Decision of Optimal Cluster(s)

At this point, I examined the relationship between the hop count and RTT and explain how to use the relationship for selection of the optimal cluster.

The RTT (latency) is the total time for propagation delay, transmission time and processing time which means in theory, a more hop counts cause more delays for a packet in theory. As is mentioned in the statistical results of CAIDA [7], while the number of hops is increasing, the range and median RTT values incrementally increase. Thus, the first decision criteria is the relation between latency and hop count. It is assumed that, a greater number of hop count means the packets have larger RTT values.

Using this information I generate a decision rule for the two attributes. The rule is as follows;

- case1: both the latency and the hop count for the packet is large - expected
- case2: both the latency and the hop count for the packet is small - expected
- case3: the latency is small but the hop count is large, it is concluded that the processing and progressing time is less because of having larger bandwidth and also it may be interpreted that the location of hops that the packet travels are closer to each other and the travel time is less for that reason. Also, CAIDA provides a statistical result that shows the positive relation between RTT and geographical distances [6]. In summary, the performance is better for those packets.
- case4: the latency is large, but the hop count is small, it can be considered that either the bandwidth is low and it causes a larger propagation delay or the hops are far from each other and the transmission time is large for the packets. Other than these two cases, there might be any unexpected reason for having larger latency, even having a smaller number of hops. It is concluded that the network performance is lower in this case.

Regarding these defined rules, case3 dominates all other cases. Case 1 and case2 might be evaluated as equal. However, I am giving priority to case 2 where the packets

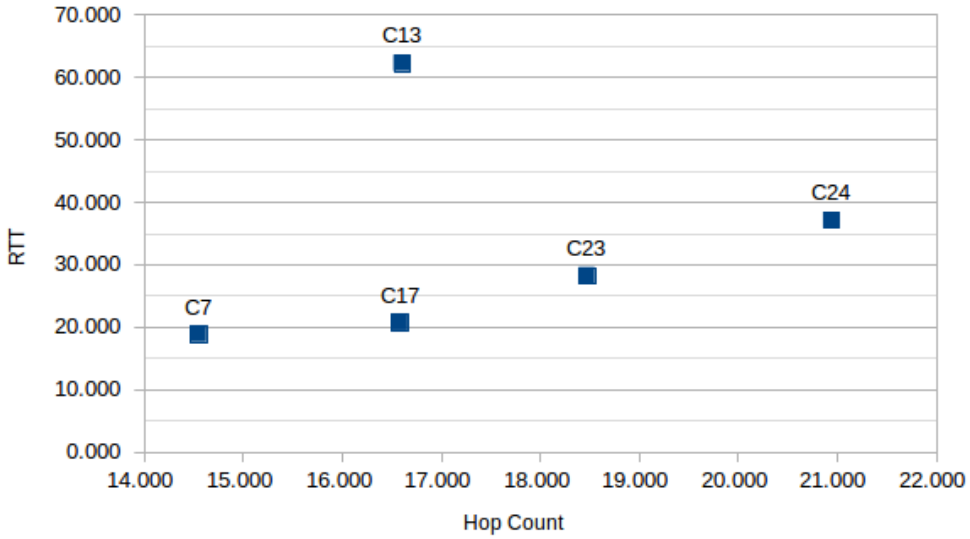

Figure 4.10: Asia Pacific 2012 Latency - Hop Count Graph

have smaller latency in expected cases. Case4 cannot dominate to other cases. To generalize the logic, the formula 4.3 helps to order the latency-hop count related network performance of clusters;

$$TPH = Latency/Hop\_count \tag{4.3}$$

where TPH refers to the ratio of total delay time of a packet to the number of hops and stands for Time Per Hop, $Latency$ is the total round trip time for a packet and Hop_count is the total number of hops that a packet needs to travel until reaching to the destination. If formula 4.3 is applied to SOM 2012 candidate clusters, the ratios will be as follows;

- C7 - 18.872 / 14.549 = 1.29
- C13 - 62.249 / 16.603 = 3.75
- C17 - 20.713 / 16.582 = 1.24
- C23 - 28.249 / 18.475 = 1.53
- C24 - 37.111 / 20.944 = 1.77

Cluster13 has the maximum ratio result which means while the RTT is higher, the packets have a lower hop count which is defined as the number of hops that a packet takes in going from the source to the destination. cluster17 has the minimum ratio value which means that although the packets travel a greater number of hops, it takes less time in comparison with the other clusters. As a result, selecting cluster17 as an optimal cluster is more reasonable with respect to hop count and RTT attributes.

Effect of Hop Count, TTL and Failure on Decision of the Optimal Cluster

The TTL value should be defined carefully since TTL / hop limit avoids undeliverable packets stuck in routing loops. On the other hand, if the TTL is defined as less than desired, the packets will be discarded before reaching the destination.

Failure has 4 different statuses: LOOP, GAPLIMIT, UNREACHED and COMPLETED. I need to discover any pattern or relationship in the dataset in order to understand the reason for each failure status.

As is mentioned above, the decision of the optimal cluster(s) can be done by taking into consideration an attribute individually, such as RTT. The packets that have a lower RTT value always have more chance to be selected as optimal. However, it is not always the case that an attribute can be considered independently for a network packet. For instance, TTL affects the failure of a packet and it is not possible to decide whether the exact TTL value should be maximized or minimized without the information of how much the packet needs to travel to reach the destination. In other words, the relational function for the TTL value with another attribute is required to be defined. That is why, a rule is defined that reveals the relation between TTL, hop count and failure which is shown in table 4.13. Therefore, the second decision criteria is the reason for the failure status of the probes regarding the relation between TTL, hop count and failure attributes. A correlation test discovers these relations. Table 4.13 shows the correlation coefficient results for defined attributes for all datasets.

As you can see, there is a weak correlation between failure and probe TTL. That is why, no relationship is claimed that the change in failure is accompanied by a change in TTL. Similarly, the relationship between failure and probe TTL is not strong and the relation can not be used directly for the decision. Although, the probe TTL and hop count do not directly affect the failure status, the difference of probe

Table 4.13: Correlation Results for Specified Attributes of SOM clustering results

Year	Location	Attributes	Correlation Coeff
2012	Asia P.	Hop Count - Failure	0.1469267
		Probe TTL - Failure	0.1065877
		Hop Count - Probe TTL	0.1273825
		Diff - Failure	0.8412736
	Europe	Hop Count - Failure	0.2839349
		Probe TTL - Failure	0.1762374
		Hop Count - Probe TTL	0.1318037
		Diff - Failure	0.9601689
	N. America	Hop Count - Failure	0.3272741
		Probe TTL - Failure	0.1977540
		Hop Count - Probe TTL	0.1069281
		Diff - Failure	0.9095368
2013	Asia P.	Hop Count - Failure	0.1914977
		Probe TTL - Failure	1.1045620
		Hop Count - Probe TTL	0.1544113
		Diff - Failure	0.9084986
	Europe	Hop Count - Failure	0.2989833
		Probe TTL - Failure	0.0780303
		Hop Count - Probe TTL	0.3350155
		Diff - Failure	0.9540549
	N. America	Hop Count - Failure	0.3557487
		Probe TTL - Failure	0.1113637
		Hop Count - Probe TTL	0.2111955
		Diff - Failure	0.9487247
2014	Asia P.	Hop Count - Failure	0.1349593
		Probe TTL - Failure	0.1858194
		Hop Count - Probe TTL	0.0944010
		Diff - Failure	0.9132981
	Europe	Hop Count - Failure	0.2770716
		Probe TTL - Failure	0.1232906
		Hop Count - Probe TTL	0.1335457
		Diff - Failure	0.9344971
	N. America	Hop Count - Failure	0.2361179
		Probe TTL - Failure	0.2837357
		Hop Count - Probe TTL	0.3001534
		Diff - Failure	0.8368203

TTL and hop count has strong correlation with failure. Instead of using hop count or probe TTL directly, the difference of hop count and TTL values is used as a criteria. It should be noted here, the difference between hop count and probe TTL will be called average failure rate (ASR), because the difference indicates the success-failure situation of probes and hop count and probe TTL scores are the average scores of all instances grouped into a cluster. Here are the patterns found in the dataset;

- If the hop count and the probe TTL are equal, the failure status is COMPLETED successfully

- If the result of ASR is equal to 5, the failure status is GAPLIMIT. Thus, it can be claimed that the gap limit is defined as 5 for the dataset.

- If the ASR value is a value between 1-5, the failure status is either UNREACHED or LOOP.

The cases mentioned above will be used for selecting the optimal cluster by using the attributes TTL, hop count and failure. The clusters that have more instances which have equal TTL and hop count values are more likely to be selected as optimal. It should be noted here, the average hop count and average probe TTL of clusters are used to calculate the difference instead of calculating the differences individually. In addition to this, there is no chance of a packet to be in the same cluster with an outlier, all the outliers are already grouped into another cluster(s) by definition of clustering.

Also, the ratio of COMPLETED packets to the total number of instances in a cluster was not considered because the correlation of difference values of hop count and probe TTL with the success / failure situation is already shown above. In other words, the smaller the ASR value means the greater the number of completed packets in a cluster.

It should be noted here, the reply TTL has lowest priority for decision making because the property is the least informative feature in my experiment. Because, there are weak correlations of reply TTL with the other attributes.

Decision Making

In this section, how the relations between attributes can be used for decision of optimal cluster(s) is demonstrated by considering all the attributes at the same time. The two decision criteria are summarized in tables 4.14 and 4.15 for the candidate clusters for SOM and K-means algorithms.

Interpretation of the SOM Results

Table 4.14 shows the summary results for the SOM algorithm. As is mentioned before, there are two decision criteria, the difference between hop count and probe TTL (ASR) and the ratio of RTT to hop count (TPH). The interpretation of the SOM results for Asia Pacific-2012 dataset is shown as follows;

- As you can see in table 4.14 , cluster7 has the smallest ASR value, also it has the second smallest TPH value. That means, it was the largest number of successfully delivered packets and the packets are delivered in the second best time per hop count. If any other cluster does not have better ASR and TPH values, there is a strong probability to select this cluster as optimal.
- Cluster13 not only has the largest TPH value but also has a bigger ASR value than cluster7, this cluster is eliminated and will not be selected as an optimal cluster.
- Cluster17 has the best TPH value but the ASR value is worse than cluster7. This cluster might be selected as an optimal cluster.
- Cluster23 has a worse TPH value than cluster7 and cluster17, and also it has the worst success rate until now. This cluster is eliminated and will not be selected as an optimal cluster.
- Cluster24 has the second worst TPH value and has the worst ASR value. This cluster is also eliminated.

At this point, the clusters7 and cluster17 have the best values in comparison to the other candidate clusters of the Asia Pacific dataset for the year 2012. If the system is not ultra-sensitive about the latency, there is no considerable difference

Table 4.14: Summary table for SOM algorithm

Year	Location	Cluster No	Hop Count	RTT	Probe TTL	TPH	ASR
2012	Asia P.	7	14.549	18.872	13.006	1.297	1.543
		13	16.603	62.249	14.003	3.749	2.600
		17	16.582	20.713	14.004	1.249	2.579
		23	18.475	28.249	14.983	1.529	3.493
		24	20.944	37.111	15.986	1.772	4.959
	Europe	7	20.953	20.010	15.990	0.955	4.963
		13	18.515	16.641	14.990	0.899	3.524
		18	16.440	11.871	14.029	0.722	2.411
		24	14.689	11.742	13.014	0.799	1.675
	N. America	4	18.625	32.465	14.972	1.743	3.653
		8	16.513	15.537	13.997	0.941	2.516
		13	14.538	20.086	13.008	1.382	1.530
		24	10.627	7.007	9.007	0.659	1.621
		25	9.784	12.106	9.036	1.237	0.747
2013	Asia P.	2	10.448	26.050	9.042	2.493	1.406
		7	12.580	43.792	12.008	3.481	0.572
		13	16.668	61.279	14.009	3.676	2.660
		17	16.505	18.156	13.870	1.100	2.634
		23	18.714	23.611	14.976	1.262	3.737
		24	21.011	31.709	16.008	1.509	5.003
	Europe	7	20.934	29.947	15.981	1.431	4.952
		13	18.534	26.603	15.016	1.435	3.519
		18	16.450	22.482	14.036	1.367	2.414
		24	14.574	16.375	13.017	1.124	1.556
		25	12.668	11.428	12.009	0.902	0.659
	N. America	4	18.578	27.369	14.965	1.473	3.613
		8	16.561	14.061	13.979	0.849	2.582
		13	14.700	16.182	13.007	1.101	1.694
		17	12.694	13.758	12.005	1.084	0.690
		22	10.359	7.270	9.020	0.702	1.338
2014	Asia P.	2	10.784	20.054	9.176	1.860	1.608
		6	12.710	29.395	12.005	2.313	0.705
		7	14.601	40.473	13.015	2.772	1.586
		13	16.724	38.773	14.007	2.318	2.717
		18	18.632	41.625	14.971	2.234	3.661
		24	21.024	48.769	15.978	2.320	5.046
	Europe	4	21.038	27.879	16.007	1.325	5.032
		5	24.365	37.679	17.340	1.546	7.025
		8	18.383	22.388	15.033	1.218	3.351
		13	16.587	21.071	13.997	1.270	2.590
		17	14.506	16.752	13.018	1.155	1.488
		21	10.038	5.420	9.491	0.540	0.547
		22	12.610	11.806	12.013	0.936	0.598
	N. America	6	13.327	22.809	12.201	1.711	1.126
		7	14.687	15.990	13.009	1.089	1.678
		8	16.639	23.927	13.970	1.438	2.669
		9	21.288	30.830	16.113	1.448	5.174
		11	10.269	8.853	9.043	0.862	1.227
		12	12.548	14.610	12.020	1.164	0.528

between TPH values for clusters 7 and 17. For that reason, the ASR value will be considered for the selection. Cluster7 has a larger number of successfully delivered packets since the ASR value is less than cluster17. Thus cluster7 is selected as the optimal cluster. Instead of selecting the optimal location(s), the selection of optimal cluster(s) are shown up to this point. In order to define the optimal location(s), the majority classes need to be found in the selected optimal cluster. The majority class indicates the class that has the largest number of instances in a cluster and can be found in table 4.1 for the Asia Pacific-2012 dataset. Class 1, 2 and 3 correspond to the locations South Korea, Sydney and Tokyo respectively for this dataset. The majority class for cluster7 is class3, which means the majority location is Tokyo. For the SOM application, the majority class can easily be found in figure 4.1 and the table is not always required. Since, the color of cluster7 is blue-ish and blue corresponds to class3, the optimal location is Tokyo for the 2012 Asia Pacific dataset, with respect to the defined decision criteria and the SOM application.

In the same manner, the candidate clusters are examined for the Europe 2012 dataset with respect to the SOM clustering results. All the TPH values are close to each other so the ASR value will be crucial for the optimal cluster decision. Cluster24 is selected because it has the smallest ASR value. Class 1, 2 and 3 correspond to the locations Ireland, Germany and the United Kingdom, respectively. The data amounts can be found in table 4.2 or in figure 4.2. As you can see, class1 has 56.387 , class2 has 67.659 and the class3 has 55.434 instances in the cluster. Although the number of instances of any class do not have an edge over the other classes, the class that has the most number of instances will be defined as the majority class. Therefore, the majority class for the optimal cluster is class2 that corresponds to the location Germany.

Cluster24 and cluster25 have better solutions than the other candidate clusters for the North America 2012 dataset. Although, cluster24 has better TPH value than cluster25, the ASR value is better for cluster25. The decision for this situation is dependent on priorities. In other words, if the latency is more crucial than the success rate for a system, cluster24 should be selected because it has a lower TPH value. However, if the success rates of the packets are the most important criteria for the system, ASR value will be more critical in the decision and cluster25 would be

the optimal cluster. Classes 1, 2 and 3 correspond to the locations San Jose, Ottawa and Virginia, respectively. The majority classes for cluster24 and cluster25 are class1 and class3, respectively. In short, the optimal locations for North America are San Jose or Virginia.

In order to compare the optimal locations for different years, The same analysis is done for the years 2013 and 2014. The optimal locations for Asia Pacific are Sydney or Tokyo for the year 2013 and Sydney for 2014. The optimal location is Germany in 2013 and Ireland in 2014 for the Europe dataset. Likewise, San Jose is selected as the optimal location in 2013 and 2014 for the North America dataset.

The optimal locations for selected continents are found as above. Also, the overall optimal location can be found. However there might be other factors which are dependent on different geographical locations and those factors may affect the network performance. For instance, all latency per hop count values for the Asia Pacific dataset are more than the other two continents. However, If these factors are ignored, Virginia or San Jose are the optimal locations for all the selected locations.

Interpretation of the K-Means Results

The analysis of the result of the K-means algorithm for the Asia Pacific-2012 dataset:

- The first cluster has the smallest ASR value, it also has one of the smallest TPH values. That means, this cluster has the largest number of successfully delivered packets and the packets have a reasonable delivery time ratio per hop count. If there is no cluster better than this one, this cluster is more likely to be selected as optimal.

- Cluster3 has a similar TPH value as cluster1 but the ASR value is smaller which indicates this cluster is eliminated and will not be selected as the optimal cluster.

- Cluster11 has a similar TPH value to cluster1 and cluster3 but this cluster has the lowest success rate. This cluster is eliminated and will not be selected as the optimal cluster.

- Cluster18 has the worst TPH value, and also it has a worse success rate than

Table 4.15: Summary table for K-means algorithm

Year	Location	Cluster No	Hop Count	Probe TTL	RTT	ASR	TPH
2012	Asia P.	1	11.601	11.591	27.73	0.01	2.390
		3	17.000	13.974	39.899	3.026	2.347
		11	21.456	15.955	51.204	5.501	2.386
		18	13.517	10.401	94.422	3.116	6.985
		20	14.472	12.986	35.449	1.486	2.449
		21	17.999	15.002	40.936	2.997	2.274
		23	15.228	13.341	76.054	1.887	4.994
	Europe	2	16.457	13.951	20.439	2.506	1.242
		4	19.507	15.430	22.244	4.077	1.140
		6	14.672	13.001	16.298	1.671	1.111
		10	21.444	15.944	30.754	5.5	1.434
	N. America	2	13.011	12.001	18.024	1.01	1.385
		6	17.467	14.384	33.082	3.083	1.894
		7	11.999	11.886	13.231	0.113	1.103
		14	13.999	12.955	20.809	1.044	1.486
		15	15.570	13.554	26.541	2.016	1.705
		20	10.106	10.000	8.131	0.106	0.805
2013	Asia P.	2	17.000	13.936	43.244	3.064	2.544
		5	15.000	12.989	40.628	2.011	2.709
		18	11.575	11.564	31.752	0.011	2.743
		21	15.410	13.870	76.252	1.54	4.948
	Europe	2	15.000	12.989	18.165	2.011	1.211
		11	19.276	19.123	64.373	0.153	3.340
		20	20.456	15.826	31.231	4.63	1.527
		22	12.544	9.023	48.541	3.521	3.870
		24	23.154	16.674	38.949	6.48	1.682
	N. America	5	9.612	6.999	8.339	2.613	0.868
		6	16.000	13.822	25.575	2.178	1.598
		8	17.000	13.926	28.951	3.074	1.703
		9	15.000	13.926	21.514	1.074	1.434
		13	18.406	14.797	30.049	3.609	1.633
		14	14.000	12.869	21.843	1.131	1.560
		16	20.856	15.816	32.493	5.04	1.558
		23	12.576	10.849	15.508	1.727	1.233
2014	Asia P.	4	9.696	8.802	16.837	0.894	1.736
		10	13.132	12.310	30.474	0.822	2.321
		17	15.541	13.437	37.002	2.104	2.381
		18	18.000	14.633	43.589	3.367	2.422
		21	17.000	13.910	42.114	3.09	2.477
		23	17.245	15.062	73.778	2.183	4.278
	Europe	3	22.439	16.241	38.138	6.198	1.700
		4	16.512	13.985	21.128	2.527	1.280
		20	13.580	12.491	14.337	1.089	1.056
		22	10.573	9.133	8.154	1.44	0.771
	N. America	1	10.464	5.989	54.034	4.475	5.164
		7	13.000	11.979	22.133	1.021	1.703
		14	20.472	15.652	31.494	4.82	1.538
		16	17.452	14.213	26.544	3.239	1.521
		20	14.477	12.862	20.542	1.615	1.419

cluster1 and cluster3. This cluster is eliminated and will not be selected as the optimal cluster.

- Cluster20 has a similar TPH with clusters1, clusters3 and clusters11 and has a better ASR value than other clusters, except cluster1. This cluster is the second cluster that can be selected as the optimal cluster.

- Lastly, although the 23rd cluster has a reasonably good success rate, the TPH value is not good enough to be selected.

Cluster1 has the best values in comparison to the other candidate clusters for the Asia Pacific dataset for the year 2012. Table 4.10 shows the data amounts for each class and those amounts will be used to find the majority class / location. Classes 1, 2 and 3 correspond to the locations South Korea, Sydney and Tokyo respectively for the Asia Pacific dataset as in the SOM clustering. The majority class for cluster1 is class2 which shows the optimal location is Sydney.

Similarly, cluster6 has the best values to be selected as the optimal cluster for the Europe 2012 dataset. Classes 1, 2 and 3 correspond to the locations Ireland, Germany and the United Kingdom, respectively. It should be noted here, the RTT and hop count values of cluster6 are the highest out of all other candidates. That is to say, considering RTT values individually will cause the number of hops the packet travelled to be ignored and the clusters not to be selected as optimal. However defining the relational decision criteria will prevent the type 1 error which indicates to reject the true result. The optimal location for the Europe dataset is Ireland.

Lastly, cluster20 has the best values in the North America dataset in 2012. Classeses 1, 2 and 3 correspond to the locations San Jose, Ottawa and Virginia, respectively. The optimal location for North America is San Jose. Also, the overall optimal location is Virginia in all the selected locations in 2012.

The optimal location for Asia Pacific for the year 2013 is Sydney as in 2012. There is no change for 3 different years in a row. The decision of the optimal location of the Europe dataset for the year 2013 is not easy, as for 2012, since The cluster has the best TPH value but the success rate is not as good as cluster11. At that point, the decision should be made according to priority. If the reliability of the packets has a higher priority for the a system, the cluster which has a greater success rate should

be selected as the optimal. However, if the delay time for the packets is the most important criteria for a system, latency / hop count value would be more influence on the decision. Thus there are two optimal clusters for this dataset, cluster2 and cluster11. The corresponding optimal locations are Ireland and the United Kingdom. If cluster2 is selected the optimal location will not change in 1 year, otherwise the optimal location will change from Ireland to the United Kingdom. There is no change for the North America dataset.

For the year 2014, the optimal location for the Asia Pacific datasetis again Sydney and the optimal location for Europe is Ireland. The same situation as the Europe dataset in 2013 occurs for the North America dataset in 2014. There are two optimal cluster regarding the priority decision, San Jose and Virginia.

The overall optimal location for the year 2013 is San Jose and the criteria values for the year 2014 are close to each other which makes it the decision dependent to the priorities. As you can see, decreasing the number of decision criteria makes easier to select the optimal location.

Comparison of SOM and K-means Results

In total, there are nine different datasets from three different main locations for three different years. The optimal locations are found for each dataset by applying SOM and K-means algorithms. As is seen in table 4.16, the optimal locations are different for SOM and K-means algorithms for the datasets employed, because of the algorithmic difference of those algorithms. Also, the parameters such as initial values, number of clusters, dataset, etc. affect the accuracy of the algorithms. The choice of the clustering algorithm is crucial and dependent on the selected parameters and the datasets employed. Osame et al. emphasized that the results of SOM generally have higher accuracy than K-means in terms of clustering [9]. Although many other studies find out there are more factors that may affect the performance of clustering algorithms, the current comparisons will be taken into consideration. In this experiment, the main purpose is to reveal an optimization method using the clustering algorithms, instead of comparing the accuracy and performance of the clustering algorithms, so the SOM results will be considered as more accurate than K-means results due to the paper [31]. However, the choice of clustering algorithm may differ in the future

regarding further studies on the performance of clustering algorithms or the employed dataset.

In this thesis, the optimal location information aims to use current data in order to predict the future. However, the results may differ due to the change of network infrastructure or the intensity of network traffic. In other words, if a location is selected as optimal to provide service and / or to deploy a new server is located due to the result, after a certain period of time the location may not remain optimal location. Thus, the measurement should be repeated to reveal either the location still has optimal network properties or the location is not optimal anymore. It should be noted here, the network properties used for this experiment, TTL and hop count do not change according to different time periods. Because TTL is a pre-defined attribute for a network and does not change during the transactions. Also the hop count is related to the network routing algorithm or strategy and does not change frequently. However the properties failure and RTT may change regarding the intensity of the network packets or the ability of the current network infrastructure. There might be different strategies in the case of different optimal location results for different times. One of the strategies is to apply the method more frequently and find out the trend in order to decide re-location or make users use the new optimal location more. Another strategy is immediately to make the users use the new optimal location or re-locate the server in the case of changing the optimal location. However, as you can see in the result table, the optimal location is completely changed in nine different datasets only twice, which are the Europe dataset from the year 2013 to 2014 and Asia Pacific dataset from 2013 to 2014 and the change is only for the SOM results. This means, if the servers are located with respect to the study results of SOM, reconfiguration / relocation is required only twice for nine different locations in three years. If the K-means results are considered, the determined optimal locations will remain as optimal in three years and there is no need to relocate the servers or direct the users to other servers. In addition, online learning approach can be applied to reveal the optimal location to make a sequence of accurate predictions which is explained in the paper [16] aiming to minimize total electrical power losses and also mentioned that the problem can be easily configured as multi-objective.

As is mentioned in the paper [9], the criteria to compare clustering algorithms

Table 4.16: Optimal Locations

Year	Location	Optimal Location for SOM	Optimal Location for K-means
2012	Asia Pacific	Tokyo	Sydney
	Europe	Germany	Ireland
	North America	San Jose OR Virginia	San Jose
2013	Asia Pacific	Tokyo OR Sydney	Sydney
	Europe	Germany	England & Ireland
	North America	San Jose	San Jose
2014	Asia Pacific	Sydney	Sydney
	Europe	Ireland & England	Ireland
	North America	San Jose	San Jose OR Virginia

are the size of the dataset, the type of the data, the number of the clusters and the sum of square errors (SSE) and SOM was found better. In addition, Thahira et al. compare the SOM and K-means algorithms in [19] and found that SSE is less for SOM compared to the K-Means clustering algorithm in most of the cases and the inter-cluster distances are more and intra cluster distances are less of SOM than K-means.

Chapter 5

Conclusion

In this thesis, I aimed to reveal a novel approach to explore the optimal location(s) for service deployment on the cloud using unsupervised algorithms. Even though the capacity and ability on computational systems are increasing day by day, the number of users and computational complexity are also exponentially increasing. Therefore, either a greater number of services / servers should be used or the existing servers should be used in a more efficient way. Locating services / servers in an optimal location with respect to network properties provides to serve a larger number of users with more efficient network performance. The problem of locating services in optimal locations addresses not only the cost but also the efficiency of the services because using the current services efficiently decreases the necessity of using more services.

To study this, I employed datasets from three main locations, namely Asia Pacific, Europe and North America, and over three different time periods, namely 2012, 2013 and 2014 provided by CAIDA. These datasets profile a general overview of the traffic at time they were captured. Thus, my aim was to discover how the network properties affect decisions of optimal location while deploying services and how it changes (if at all) over time. Moreover, to the best of my knowledge this is the first time where these network properties has been used for service location optimization using the SOM and K-means unsupervised learning algorithms in order to cluster the network data. The reasons for selecting these algorithms: being the most commonly used algorithms, being easy to implement and the ability of handling high dimensionality [9]. The results of the experiments are summarized as follows:

- The network characteristics have significant effect on service location optimization. The results show that, the selected optimal locations remain as the optimal location(s) for the following three years for seven of nine different datasets. That means, the locations that remain as optimal for the following years still have the optimal network property results and in this period of time the network

performance is better than other locations.

- This method is eligible to comply with new developments. For example, it is possible to apply the new solutions on clustering to the proposed methodology.

- The method represents a low-dimensional result that is easy to understand and interpret. Especially, if SOM is used as a clustering method, the 2-D mapping visualization results will make the decision easier. The decision of possible optimal locations can be done using the 2-D mapping representation without requiring more computation or analysis.

- Using Machine Learning algorithms provide self-learning that seems to improve the results for the future. The method is able to find out the changes of the network properties for a specific location or time. Especially, if there is an abrupt change on the network properties, the method can be applied more frequently and it will help to understand whether the change is temporary or not.

- The method is eligible to work with big data regardless of the amount of data and is able to provide low-dimensional, abstract output. Thus, the most powerful feature of the method is the abstraction of the data and reducing the dimensionality. Regardless of the data-size, the suggested systems and algorithms are able to apply the process.

  .

In order to create the decision rules after clustering, the patterns or correlations in the datasets are discovered. Instead of using each network property individually, the discovered relations are used. Although the method is evaluated on an off-line datasets, the system can be integrated to an on-demand system and streaming data can also be used. The method could help to minimize the network performance problems because the network properties help to reveal the problems that could occur on the network and suggest a better location deploy a service.

Future work will explore Genetic Algorithms for optimization in the same manner. In addition, trend analysis can be applied to the results of the method. If the method

is applied on a regular basis, for trend analysis, the predictions become more reliable and accurate.

Bibliography

[1] *Internet usage statistics.* http://www.internetlivestats.com/internet-users, 2006.

[2] *Design Best Practices for Latency Optimization.* CISCO, 2007.

[3] *Human interface and the management of information.* 16th International Conference, HCI international, Heraklion, Crete, Greece, 2014.

[4] *JSON.* https://tools.ietf.org/html/rfc7159.html, 2014.

[5] *Archipelago Monitor Locations.* https://www.caida.org/projects/ark/locations/map, 2017.

[6] *The relation between RTT and Geographical distance provided by CAIDA.* https://www.caida.org/projects/ark/statistics/aal-dk/med_rtt_vs_dist.html, 2017.

[7] *The relation between RTT and number of hops.* https://www.caida.org/projects/ark/statistics/dub-ie/med_rtt_per_hop.html, 2017.

[8] *Scamper.* http://www.caida.org/tools/measurement/scamper/, 2017.

[9] Osame Abu Abbas. *Comparison between data clustering algorithms.* The International Arab Journal of Information Technology, 2008.

[10] Oded Berman and Zvi Drezner. *The multiple server location problem.* Journal of the Operational Research Society 58.1: 91-99., 2007.

[11] Richard C. Larson Berman, Oded and Samuel S. Chiu. *Optimal server location on a network operating as an M/G/1 queue.* Operations research 33.4:746-771, 1985.

[12] Ab Ghani P. Zolkepali N. A. Selvarajah S. Haniff J. Bujang, M. A. *A comparison between convenience sampling versus systematic sampling in getting the true parameter in a population: explore from a clinical database: the Audit Diabetes Control Management (ADCM) registry in 2009.* Statistics in Science, Business, and Engineering (ICSSBE), 2012 International Conference on. IEEE, 2012.

[13] YuFeng Deng and Sathiamoorthy Manoharan. *A review of network latency optimization techniques.* Communications, Computers and Signal Processing (PACRIM), 2013 IEEE Pacific Rim Conference on. IEEE, 2013.

[14] Jitendra Padhye Draves, Richard and Brian Zill. *Comparison of routing metrics for static multi-hop wireless networks.* ACM SIGCOMM Computer Communication Review. Vol. 34. No. 4. ACM, 2004.

[15] Peris V. Saha D. Basturk E. Haas R. Engel, R. *Using IP anycast for load distribution and server location.* Proc. of IEEE Globecom Global Internet Mini Conference, 1998.

[16] Francis R. Bach Hoffman, Matthew and David M. Blei. *Online learning for latent dirichlet allocation.* Advances in neural information processing systems, 2010.

[17] Anil K. Jain. *Data Clustering: 50 Years Beyond K-Means.* Pattern recognition letters 31.8, 2010.

[18] Gihun Jung and Kwang Mong Sim. *Location-aware dynamic resource allocation model for Cloud computing environment.* International Conference on Information and Computer Applications, IPCSIT. Vol. 24., 2012.

[19] Sameema Zahid Ansari Kadijath Thahira, Jovita Vani Sequeira. *Performance Analysis of Self-Organizing Neural Network-Based Clustering.* International Journal of Advanced Research in Computer Science and Software Engineering, 2015.

[20] Masaki Kohana and Shusuke Okamoto. *A Server Selection Method for Web-Based Multiserver Systems.* 2015 IEEE 29th International Conference on Advanced Information Networking and Applications. IEEE, 2015.

[21] Teuvo Kohonen. *MATLAB Implementations and Applications of the Self-Organizing Map.* Unigrafia Oy, 2014.

[22] Robert V. Krejcie and Daryle W. Morgan. *Determining sample size for research activities.* Educational and psychological measurement 30.3 (1970): 607-610., 1970.

[23] Federico Larumbe and Brunilde Sans. *Optimal location of data centers and software components in cloud computing network design.* Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing(ccgrid), IEEE Computer Society, 2012.

[24] Ben Liang and Zygmunt J. Haas. *Comparison between data clustering algorithms.* Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE., 2003.

[25] Matthew Luckie. *Scamper: a Scalable and Extensible Packet Prober for Active Measurement of the Internet.* IMC '10 Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, 2010.

[26] K. Claffy Nevil Brownlee. *Internet measurement and data analysis:topology, workload, performance and routing statistics.* NAE Workshop, 1999.

[27] Ramesh K. Sitaraman Nygren, Erik and Jennifer Sun. *The Akamai network: a platform for high-performance internet applications.* ACM SIGOPS Operating Systems Review 44.3:2-19, 2010.

[28] Ammar Mostafa H. Fei Zongming Bhattacharjee Samrat NZegura, Ellen W. *Application-layer anycasting: a server selection architecture and use in a replicated Web service.* IEEE/ACM Transactions on Networking 8.4: 455-466, 2000.

[29] Katia Obraczka and Fabio Silva. *Network latency metrics for server proximity.* Global Telecommunications Conference, 2000. GLOBECOM'00. IEEE. Vol. 1. IEEE, 2000.

[30] Quang-My Tran and Arek Dadej. *Optimizing cached route Time-To-Live in mobile ad-hoc networks.* T2015 IEEE 29th International Conference on Advanced Information Networking and Applications. IEEE, 2015.

[31] Iyer Aurobind Venkatkumar and Sanatkumar Jayantibhai Kondhol Shardaben. *Comparative study of data mining clustering algorithms.* Data Science and Engineering (ICDSE), 2016.

[32] Huang C. Li J. Ross K. W. Wang, Y. A. *Estimating the performance of hypothetical cloud service deployments: A measurement-based approach.* INFOCOM, Proceedings IEEE. IEEE, 2011.